

1989

Optimal system reliability design of consecutive-k-out-of-n systems

Mingjian Zuo
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>



Part of the [Industrial Engineering Commons](#), and the [Systems Engineering Commons](#)

Recommended Citation

Zuo, Mingjian, "Optimal system reliability design of consecutive-k-out-of-n systems " (1989). *Retrospective Theses and Dissertations*. 11178.
<https://lib.dr.iastate.edu/rtd/11178>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

90

14973

UMI

MICROFILMED 1990

INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9014978

**Optimal system reliability design of consecutive- k -out-of- n
systems**

Zuo, Mingjian, Ph.D.

Iowa State University, 1989

U·M·I

**300 N. Zeeb Rd.
Ann Arbor, MI 48106**

**Optimal system reliability design of
consecutive- k -out-of- n systems**

by

Mingjian Zuo

**A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY**

Major: Industrial Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

**Iowa State University
Ames, Iowa
1989**

TABLE OF CONTENTS

ACKNOWLEDGMENTS	viii
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. REVIEW OF k-OUT-OF-n SYSTEMS	6
Assumptions, Notation, and Definitions	7
Assumptions	7
Notation	7
Definitions	9
System Reliability Evaluation	10
Inclusion-exclusion method	12
Sum of disjoint products	14
Improved inclusion-exclusion method	20
Generating function method	22
JG-1 method	23
RSPK method	25
Rushdi's method	26
Summary	31

CHAPTER 3. REVIEW OF CONSECUTIVE- k -OUT-OF- n SYS-

TEMS	32
Notation and Assumptions	32
Notation	32
Assumptions	34
The Consecutive- k -out-of- n :F Systems	34
System reliability evaluation	34
Bounds and approximations of system reliability	39
Optimal system design	45
The Consecutive- k -out-of- n :G Systems	49
Reliability and reliability bound evaluation	49
Optimal system design	51
Special Consecutive- k -out-of- n :F Systems	52
Summary	56

CHAPTER 4. ANALYSIS OF k -OUT-OF- n AND CONSECUTIVE-

k -OUT-OF- n SYSTEMS	58
k -out-of- n Systems	58
Consecutive- k -out-of- n Systems	65
Special reliability formulas	65
Disjoint minimal path method	69
Component importances	71
Strict Consecutive- k -out-of- n Systems	78

Summary	80
CHAPTER 5. OPTIMAL SYSTEM DESIGN OF CONSECUTIVE- k-OUT-OF-n SYSTEMS	81
Introduction	81
Invariant Optimal System Designs	82
Variant Optimal System Designs	95
Heuristic method	95
Randomization method	104
Binary search method	105
Summary	106
CHAPTER 6. CASE STUDIES	107
A Sea Port	107
Photograph Techniques in a Nuclear Accelerator	113
Summary	119
CHAPTER 7. SUMMARY AND CONCLUSIONS	120
BIBLIOGRAPHY	122

LIST OF TABLES

Table 2.1:	Reliability array for calculating $R(5,8)$	30
Table 3.1:	Necessary conditions of the optimal design of linear and circular consecutive- k -out-of- $n:F$ systems	48
Table 3.2:	Summary of invariant optimal designs of linear consecutive- k -out-of- $n:F$ systems	49
Table 5.1:	Invariant optimal designs of linear consecutive- k -out-of- n systems	96
Table 5.2:	Invariant optimal designs of circular consecutive- k -out-of- n systems	97
Table 5.3:	Comparison of initial solutions on optimal system design with Heuristic 1	101
Table 5.4:	Comparison of initial solutions on optimal system design with Heuristic 2	101
Table 5.5:	Test results on 100 sets of randomly generated component reliabilities in range of $(0,1)$	103
Table 5.6:	Test results on 100 sets of randomly generated component reliabilities in range of $(0.8,0.99)$	103

Table 6.1:	Berth availability importances when the berths are i.i.d. . . .	109
Table 6.2:	A set of berth availabilities	109
Table 6.3:	Berth availability importances with optimal arrangement . .	111
Table 6.4:	Top five configurations from data in Table 6.2	112
Table 6.5:	Results obtained with the randomization method	113
Table 6.6:	A set of camera reliabilities	114
Table 6.7:	Camera reliability importances with the optimal design . . .	116
Table 6.8:	The top five configurations from data in Table 6.6	117
Table 6.9:	Heuristic result for the circular consecutive- k -out-of- n :G system	118
Table 6.10:	Sub-optimal solution obtained with the randomization method	118

LIST OF FIGURES

Figure 1.1:	A linear consecutive-3-out-of-6:F system	4
Figure 1.2:	A linear consecutive-2-out-of-7:G system	5
Figure 2.1:	An example of an irrelevant component	9
Figure 2.2:	Signal flow graph for obtaining $R(3, 7)$ and $F(3, 7)$	30
Figure 4.1:	k -out-of- n :G system reliability and cost as functions of n with $p = 0.9$, $k = 3$, and $\text{Cost} = 0.045n$	63
Figure 4.2:	k -out-of- n :G system reliability as a function of p with $n = 7$, $k = 3$, and $\text{Line} = p$	64
Figure 4.3:	The pattern of component reliability importance of a consec- utive-2-out-of-20:G system with $p = 0.5$	79
Figure 6.1:	The optimal arrangement of cameras	116

ACKNOWLEDGMENTS

I wish to express my sincerest gratitude to my major professor, Dr. Way Kuo. His guidance, support, encouragement, and numerous suggestions provided an invaluable contribution to my study and this dissertation. I feel very fortunate to have such a respectable advisor. His persistence in high quality research, devotion to education, kindness, and wisdom are admired and respected.

Each of the members of the committee made valuable contribution to this research and my five years of study at Iowa State University. Thanks go to Dr. H. A. David, Dr. W. Q. Meeker, and Dr. H. T. David for helping me build a strong probability and statistics background. I thank Dr. H. A. David for his concern about my research's progress and pointing out to me different references that I missed. Dr. Meeker and Dr. H. T. David's course in Reliability provided me with insight and understanding of system reliability and optimal system design. Thanks also go to Dr. Keith L. McRoberts for his support of my study and comments on my thesis. I also express my thanks to Dr. John Jackman for being on my graduate committee.

Finally, special thanks goes to my wife, Ninghe, for her love and support. Thanks are also extended to my parents for their understanding and support.

CHAPTER 1. INTRODUCTION

Systems are becoming a prevalent feature of our society. Complicated devices such as computers and airplanes and networks like telephone networks and various electronic networks are examples of systems. These systems are built with various components to perform specified tasks. It is often impossible to assure that the systems will perform the tasks for which they are designed. Failure of components due to causes difficult to anticipate and impossible to prevent may lead to the failure of the entire system.

The importance and utility of a system depend on its performance, and its performance depends on its design. A common measure of performance is reliability. The reliability of a system is defined as the probability that it will perform the task for which it is designed. Reliability problems have become more and more important, especially for complex and high technology systems. They are particularly critical when there are concerns over the consequences of system failure in terms of safety and cost. The tragedy of the space shuttle Challenger is a good example.

Because system is a broad concept and many systems are large in size and complicated in design, it is virtually impossible to develop universal theories for general systems. Therefore, researchers have been concentrating on systems with special structures, for example, series systems, parallel systems, series-parallel systems,

parallel-series systems, k -out-of- n systems, consecutive- k -out-of- n systems, and complex systems.

Recently, there has been considerable interest in the k -out-of- n :F systems, the k -out-of- n :G systems, the consecutive- k -out-of- n :F systems, and the consecutive- k -out-of- n :G systems. A k -out-of- n :G system is good if and only if at least k of its n components are good. A k -out-of- n :F system fails if and only if at least k of its n components fail. A consecutive- k -out-of- n :F system is a sequence of n ordered components such that the system works if and only if less than k consecutive components fail. A consecutive- k -out-of- n :G system consists of an ordered sequence of n components such that the system works if and only if at least k consecutive components in the system are good. The consecutive- k -out-of- n systems are further divided into linear systems and circular systems corresponding to the cases that the components are ordered along a line and a circle, respectively. Many researchers have focused on these special systems mainly because (1) such systems are more general than pure parallel or pure series systems, (2) some interconnection networks [2] can be handled using this technique, and (3) they are frequently encountered in practice.

k -out-of- n :F and consecutive- k -out-of- n :F systems become series systems when $k = 1$ and parallel systems when $k = n$. k -out-of- n :G systems and consecutive- k -out-of- n :G systems become series systems when $k = n$ and parallel systems when $k = 1$. A k -out-of- n :F system is equivalent to an $(n - k)$ -out-of- n :G system.

An airplane with four engines can be modeled as a 3-out-of-4:G system. It could happen that three out of four engines operational during a certain flight would not be disastrous but the loss of one more engine would be [7]. As another example,

consider a large truck equipped with eight tires. This is an example of a 4-out-of-8:G system. Although the system performance may be degraded if less than eight tires are operational, rearrangement of the tire configuration will result in adequate performance as long as at least four tires are operational [39]. In a communication network, it may be necessary to have at least k nodes operational to keep the network connected.

There exist many applications of the consecutive- k -out-of- n :F systems. One example is the system of street lights. If less than k consecutive lights are out, this system with n lights has not failed to light the way adequately [16]. A telecommunications system with n relay stations is also a good application of such systems. Suppose that the stations numbered consecutively from 1 to n are lined up and the signal transmitted from a station is only strong enough to reach the next k stations. Therefore, the signal relayed will be interrupted if and only if at least k consecutive stations fail [19]. Another example of consecutive- k -out-of- n :F systems is an oil pipeline system with n pump stations. Each station is powerful enough to send oil as far as to the next k stations. If less than k consecutive stations fail, the flow of oil will not be interrupted and the pipeline system will still function properly [19].

In quality control lot acceptance sampling, a consecutive- k -out-of- n system concept is also applicable. If consecutive k out of n lots are rejected under normal sampling scheme, tightened sampling scheme becomes effective. If consecutive k out of n lots are accepted under tightened sampling, normal sampling returns. If consecutive k out of n lots are rejected under tightened sampling, inspection is discontinued and the products are rejected.

An example of the consecutive- k -out-of- n :G system is a railway station of n lines. Because of some particular requirements, a special train can enter the station only if at least k consecutive lines are available [48].

The configuration of a linear consecutive-3-out-of-6:F system is given in Figure 1.1. An example of the consecutive- k -out-of- n :G system is depicted in Figure 1.2. It is a linear consecutive-2-out-of-7:G system [79].

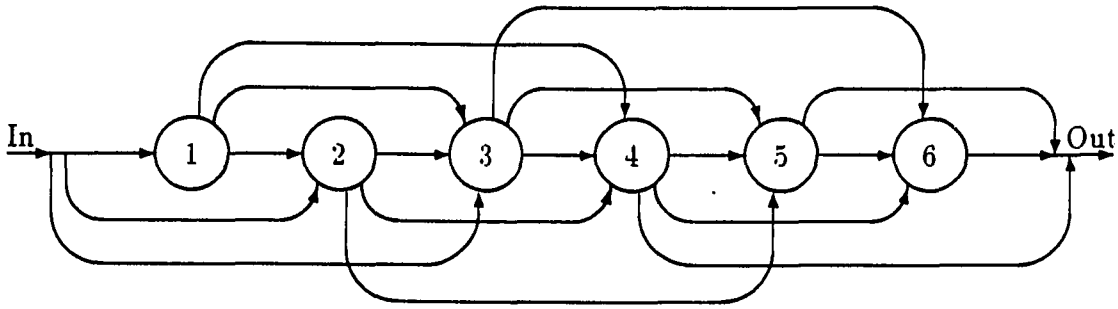


Figure 1.1: A linear consecutive-3-out-of-6:F system

This research reviews the reliability evaluation of the k -out-of- n systems and the reliability evaluation and optimal system design of the consecutive- k -out-of- n systems, investigates the properties of the k -out-of- n and consecutive- k -out-of- n systems, and then concentrates on the optimal design of the consecutive- k -out-of- n systems. Invariant optimal configurations are obtained for some consecutive- k -out-of- n systems and a heuristic method is provided for other consecutive- k -out-of- n systems where invariant optimal configurations do not exist. Case studies are provided to illustrate the applications of the theoretical results developed.

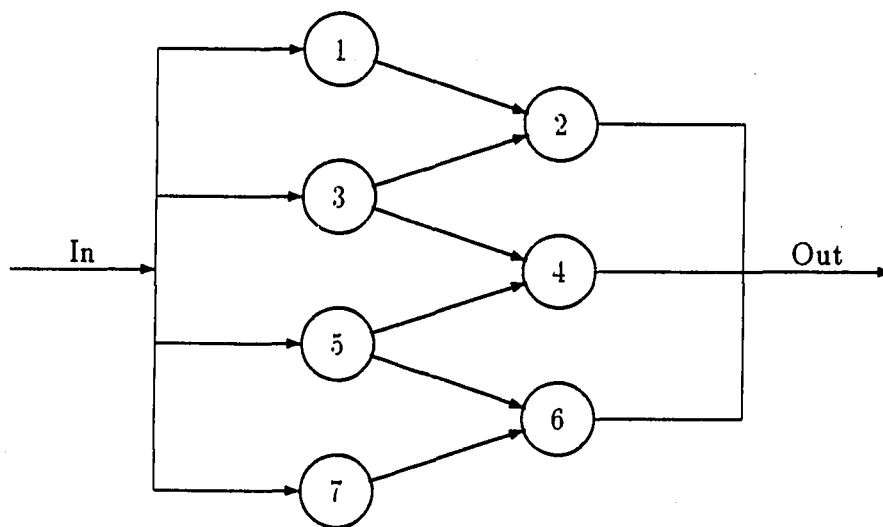


Figure 1.2: A linear consecutive-2-out-of-7:G system

CHAPTER 2. REVIEW OF k -OUT-OF- n SYSTEMS

Reliability evaluation of a system deals with computing or approximating the probability that the system functions as it is intended. The system configuration may be represented by a diagram, a list of paths, or a list of cuts. A logic function is derived from this configuration; then a probability formula is generated from the logic function. The system's reliability is obtained by substituting the component reliabilities into the formula. The classical method of generating such a formula is the inclusion-exclusion method (IE).

During the 1970s, an important development resulted in methods to derive the logic, generate the formula, and compute the reliability. In 1973, Fratta and Montanari [29] published an algorithm for the Sum of Disjoint Products (SDP), which is applicable for any system reliability evaluation problem. In 1979, Abraham [1] published an improved version of SDP. Because system reliability evaluation is very complicated, a general efficient algorithm is hard or impossible to find. In the 1980s, improvements were made to IE and SDP methods for reliability evaluations of k -out-of- n systems. Some new algorithms were also developed for k -out-of- n system reliability evaluations. This chapter reviews the developments on system reliability of the k -out-of- n systems.

Assumptions, Notation, and Definitions

Assumptions

1. Components of the system are numbered from 1 to n .
2. Each component and the system is either good or failed.
3. The state of the system is determined completely by the states of the components.
4. The components are statistically independent.

Notation

n	number of components in the system
k	minimum number of good (failed) components required for a k -out-of- n :G (k -out-of- n :F) system to be good (failed)
Φ	a null set
$+$	summation or union when the two sets are disjoint
E_i	the event that component i is good
\bar{E}_i	the event that component i is failed
p_i	$\Pr(E_i)$, reliability of component i
q_i	$\Pr(\bar{E}_i)$, unreliability of component i , $q_i = 1 - p_i$
IN_n	$\{1, 2, 3, \dots, n\}$
x_i	indicator of the state of the i th component $x_i = \begin{cases} 0 & \text{it is failed} \\ 1 & \text{it is good} \end{cases}$

X	indicator of the state of the set of components. $N_n, X = (x_1, x_2, \dots, x_n)$
m_e	an even integer number
m_o	an odd integer number
$\phi(X)$	structure function of the system $\phi(X) = \begin{cases} 0 : & \text{the system is failed} \\ 1 : & \text{the system is good} \end{cases}$
m_T	number of minimal paths in the system
$m_{C'}$	number of minimal cuts in the system
T_i	minimal path $i, i \in N_{m_T}$
C'_i	minimal cut $i, i \in N_{m_{C'}}$
T_{jD}	$T_{jD} \subset T_j, T_{jD} \cap T_i = \Phi, \text{ for } i < j$
D_i	the event that minimal path i, T_i , is good, i.e., all its components are good. $i \in N_{m_T}, D_i \equiv \bigcap_{j \in T_i} E_j$
B_i	the event that minimal cut i, C'_i , is failed, i.e., all its components are failed. $i \in N_{m_{C'}}, B_i \equiv \bigcap_{j \in C'_i} \bar{E}_j$
R_s	system reliability
U_s	system unreliability. $U_s = 1 - R_s$
R_m	estimate of R_s at step m
U_m	estimate of U_s at step m
$\ I\ $	number of elements in set I or cardinality of I
MT_j	$\{I \mid I \subset N_{m_T}, \ I\ = j\}$
MC'_j	$\{I \mid I \subset N_{m_{C'}}, \ I\ = j\}$
N_j	$\{I \mid I \subset N_n, \ I\ = j\}$

Definitions

The *structure function* ϕ of a system is a binary function of the states of the components [27].

$$\phi(X) \equiv \phi(x_1, x_2, \dots, x_n)$$

The i th component is *irrelevant* to the structure ϕ if ϕ is consistent in x_i ; that is, $\phi(1_i, X) = \phi(0_i, X)$. Otherwise, the i th component is *relevant*. Here we have used the notation:

$$(1_i, X) = (x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$$

$$(0_i, X) = (x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$$

An irrelevant component has no contribution to the system's performance. For example, component 2 is irrelevant to the structure pictured in Figure 2.1.

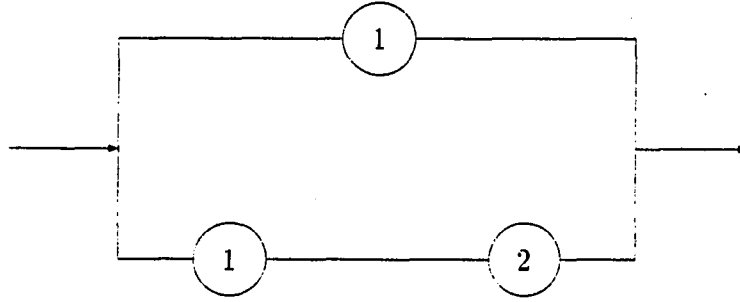


Figure 2.1: An example of an irrelevant component

A system of components is *coherent* if (a) its structure function ϕ is non-decreasing in X , i.e., $\phi(X_2) \geq \phi(X_1)$ if $X_2 \geq X_1$, and (b) each component is relevant. Let us assume that the structure (C, ϕ) is coherent. Define $C_0(X) = \{i \mid x_i = 0\}$ and $C_1(X) = \{i \mid x_i = 1\}$.

A *path vector* is a vector X such that $\phi(X) = 1$. The corresponding *path set* is $C_1(X)$. A *minimal path vector* is a path vector X such that

$$Y < X \rightarrow \phi(Y) = 0.$$

The corresponding *minimal path set* is $C_1(X)$. Physically, a minimal path set is a minimal set of elements whose functioning insures the functioning of the system.

A *cut vector* is a vector X such that $\phi(X) = 0$. The corresponding *cut set* is $C_0(X)$. A *minimal cut vector* is a cut vector X such that

$$Y > X \rightarrow \phi(Y) = 1.$$

The corresponding *minimal cut set* is $C_0(X)$. Physically, a minimal cut vector is a minimal set of elements whose failure causes the failure of the system.

In the above definitions, by $X_2 \geq X_1$, we mean $x_{2i} \geq x_{1i}$, for $i = 1, 2, \dots, n$. By $Y < X$, we mean $y_i \leq x_i$ for $i = 1, \dots, n$, with $y_i < x_i$ for at least one i .

System Reliability Evaluation

A k -out-of- n :G system is good if and only if at least k of its n components are good. It is failed if and only if less than k of its n components are good, i.e., if at least $n - k + 1$ of its n components are failed. A k -out-of- n :F system is failed if and only if at least k of its n components are failed. Therefore, that a k -out-of- n :G system is good is equivalent to that an $(n - k + 1)$ -out-of- n :F system is good, provided that these two systems have the same set of components. Similarly argued, a k -out-of- n :F system is equivalent to an $(n - k + 1)$ -out-of- n :G system. As a result, we can always

select one system instead of the other to work with in order to make the reliability evaluation more efficient.

For methods using minimal paths or minimal cuts to compute the system's reliability, we have the following arguments. A k -out-of- n :G system is good if and only if at least one minimal path is good; an $(n - k + 1)$ -out-of- n :F system is failed if and only if at least one minimal cut is failed. There are $\binom{n}{k}$ minimal paths in a k -out-of- n :G system and $\binom{n}{n-k+1}$ minimal cuts in a $(n - k + 1)$ -out-of- n :F system. The complexities for the reliability and unreliability evaluations are the functions of the number of minimal paths and the number of minimal cuts, respectively. Since a k -out-of- n :G system is equivalent to a $(n - k + 1)$ -out-of- n :F system, we can choose a system with m_T or $m_{C'}$, whichever is smaller, to work with. To find the reliability of a k -out-of- n :G system, we can use either of the following:

$$R_s = \Pr\{\text{at least one minimal path works}\} \quad (2.1)$$

$$= 1 - \Pr\{\text{at least one minimal cut fails}\}. \quad (2.2)$$

Formula (2.1) needs $m_T = \binom{n}{k}$ steps, while Formula (2.2) needs $m_{C'} = \binom{n}{n-k+1}$ steps. We can use $\min\{m_T, m_{C'}\}$ to select a formula from (2.1) and (2.2) to calculate the system's reliability.

In the following sections, we will discuss only the calculation of R_s for a k -out-of- n :G system using (2.1). If $m_{C'} < m_T$, the following substitutions may be made first,

$$\begin{aligned} m_T &\leftarrow m_{C'} \\ k &\leftarrow n - k + 1 \end{aligned}$$

$$p_i = 1 - p_i, \quad i \in N_n.$$

Then, the algorithms to be discussed will be used. After the algorithm is finished, the following determines the reliability of the original system.

$$R_s = 1 - R_s.$$

For the methods which do not use the concepts of minimal path or minimal cut, corresponding substitutions will be presented for selecting a reliability or unreliability formula such that a more efficient formula is used.

Inclusion-exclusion method

The inclusion-exclusion method (IE), also known as Poincare's theorem, is derived in the same way as statisticians calculate the probability of the union of two events: first add the probabilities of the separate events, then subtract the probability of the joint event. This results in a probability formula with alternating additive and subtractive terms.

If a system G is serial with a single path, T , the system's reliability is a single term, $\Pr\{T\}$, the probability that all the components of the path work. If we assume that there are two alternative paths, T_1 and T_2 , then the system's reliability, $\Pr\{T_1 \text{ or } T_2\}$, has three terms.

$$\Pr\{T_1 \text{ or } T_2\} = \Pr\{T_1\} + \Pr\{T_2\} - \Pr\{T_1 \text{ and } T_2\} \quad (2.3)$$

Suppose G has m_T minimal paths; m_T is any integer greater than two. The buildup of the reliability formula extends formula (2.3). The following formulas by

Heidtmann [37] allow the IE method to compute system reliability recursively.

$$R_0 \equiv 0$$

$$R_m = R_{m-1} + (-1)^{m-1} \sum_{I \subset MT_m} \Pr\{\cap_{i \in I} D_i\} \quad (2.4)$$

$$R_{m_e} \leq R_s \leq R_{m_o} \quad m_e, m_o \in N_{m_T} \quad (2.5)$$

$$R_s = R_{m_T}. \quad (2.6)$$

Formula (2.5) is referred to as Bonferroni Inequalities.

When $m_o = 1$ and the components are statistically independent, we have the following inequality from Formula (2.5):

$$R_s \leq \sum_{r=1}^{m_T} \prod_{i \in T_r} p_i \quad (2.7)$$

where T_r is the r th minimal path. This is referred to as Boole's Inequalities.

As stated in (2.5), (2.4) provides successive upper and lower bounds for system reliability. When exact system reliability is not desired, Formula (2.4) can be used to find bounds on exact system reliability. It is not true in general that the upper bounds decrease and the lower bounds increase, however, the bounds eventually do converge to exact system reliability [37].

In system reliability evaluations, generally the minimal paths or the minimal cuts have elements in common. Assume that a component E is contained in both minimal paths T_1 and T_2 . Then $\Pr\{E\}$ will appear in $\Pr\{T_1\}$, $\Pr\{T_2\}$, and $\Pr\{T_1 T_2\}$. The term in $\Pr\{T_1 T_2\}$ cancels with the term in $\Pr\{T_1\}$ or $\Pr\{T_2\}$. With the IE method, a large number of pairs of identical terms with opposite signs cancel. For any system with m_T minimal paths, each of which has a unique set of components, the number

of terms in step m ($m \leq m_T$) of the method is $\binom{m_T}{m}$. Thus, the total number of terms generated to find exact system reliability with Formula (2.4) is $\sum_{m=1}^{m_T} \binom{m_T}{m} = 2^{m_T} - 1$ and this is the number of terms generated by the IE method. When the minimal paths have components in common, which is the case for k -out-of- n systems, the actual number of terms is a fraction of this maximum number. Therefore, finding a method to avoid generating these cancelling terms affords an important computational advantage.

Sum of disjoint products

Like the IE method, the sum of disjoint products method (SDP) derives a formula which is a sum of products. Unlike IE, however, the formula is entirely additive; there are no exclusions and every term has a plus sign.

The addition law of probabilities is the underlying justification for SDP. If two or more events have no elements in common, then the probability that at least one of them occurs is the sum of the probabilities of the separate events. For example, with two events A and B , let \bar{A} denote the complement of A . Then we have

$$\Pr\{A \text{ or } B\} = \Pr\{A\} + \Pr\{\bar{A}B\}. \quad (2.8)$$

Similarly with three events, A , B , and C

$$\Pr\{A \text{ or } B \text{ or } C\} = \Pr\{A\} + \Pr\{\bar{A}B\} + \Pr\{\bar{A}\bar{B}C\}. \quad (2.9)$$

With n events A_1, A_2, \dots , and A_n :

$$\Pr\{A_1 \text{ or } A_2 \text{ or } \dots \text{ or } A_n\} \quad (2.10)$$

$$= \Pr\{A_1\} + \Pr\{\bar{A}_1 A_2\} + \dots + \Pr\{\bar{A}_1 \bar{A}_2 \dots \bar{A}_{n-1} A_n\}. \quad (2.11)$$

For a system with m_T minimal paths, T_1, T_2, \dots and T_{m_T} , the event that the system works means that at least one of the m_T paths works. Therefore,

$$\begin{aligned} \phi(\text{System}) &= T_1 \text{ or } T_2 \text{ or } \dots \text{ or } T_{m_T} \\ &= T_1 - \bar{T}_1 T_2 + \bar{T}_1 \bar{T}_2 T_3 - \dots \\ &\quad + \bar{T}_1 \bar{T}_2 \dots \bar{T}_{m_T-1} T_{m_T}. \end{aligned} \quad (2.12)$$

Even though Formula (2.12) can be used to calculate system reliability, the problem at hand is how to find an efficient way to obtain disjoint terms such that system reliability can be obtained with a simple summation of product terms.

Fratta and Montanari (F&M) presented a method to compute the exact terminal reliability given the set of all minimal paths between two nodes in a network [29]. A modification of this method was presented as well to approximate exact system reliability. It can be done symbolically by transforming a Boolean sum of products into an equivalent form in which all terms are disjoint.

Algorithm: F_0 is a sum of products and each product corresponds to a minimal path. For example, F_0 may be a sum of two product terms, $x_1 x_2$ and $x_1 x_3$, where $x_1 x_2$ corresponds to minimal path T_1 with components 1 and 2 and $x_1 x_3$ corresponds to minimal path T_2 with components 1 and 3. F_m is also a sum of products, $1 \leq m \leq m_T$. At step m of the reliability evaluation, a product term from F_{m-1} , say S , is considered:

$$R_0 = 0 \quad (2.13)$$

$$F_0 = T_1 + T_2 + \dots + T_{m_T} \quad (2.14)$$

$$R_m = R_{m-1} + \Pr\{S\} \quad (2.15)$$

$$F_m = F_{m-1} \bar{S}. \quad (2.16)$$

F_m must be a Boolean sum of product terms. The algorithm terminates when there are no product terms in F_m .

The selection of the term from F_{m-1} can be performed according to different criteria. A very good one could be to choose the one whose probability is the largest. This gives, as a result, an algorithm with the best estimate to R_s at each step but requires at each iteration the computation of the probabilities corresponding to all terms of the Boolean function F_{m-1} . Another criterion is to select the term with a minimum number of factors or a minimum number of complemented factors. The latter takes advantage of the fact that the q_i 's are always much smaller than the corresponding p_i 's in a communication network and thus, it works very often as the maximum probability criterion.

Fratta and Montanari [29] also presented a modified version of the above algorithm to estimate exact system reliability with a given error rate. At each iteration of the above algorithm, the current value R_m is increased by a positive quantity corresponding to the probability of the selected implicant. This means that at each iteration, the procedure gives an estimate with a positive error of system reliability. This error is non-negative and monotonically decreasing with m . The stopping condition is when there are no terms in F_m . In the modified version of the algorithm, the probability of F_m is evaluated at each step. If this probability is smaller than a given error, the algorithm is terminated. This probability is the upper bound for error of the estimate of system reliability because the terms in F_m are not necessarily disjoint.

F&M's algorithm of exact system reliability computation wastes time doing extra works because an F_m function is carried from step to step. However, F_m is useful in providing a lower bound on system reliability.

Aggarwal, Misra, and Gupta (AMG) [2] presented an idea similar to F&M's [29]. However, this algorithm is more efficient because the F function is abandoned and a rapid algorithm for calculating the disjoint sums is adopted.

With m_T minimal paths, T_1, T_2, \dots and T_{m_T} , Aggarwal et al. proposed a method to find the corresponding disjoint minimal paths, $T_1, T_{2D}, T_{3D}, \dots$ and T_{m_TD} . Here T_{jD} is a subset of T_j such that T_{jD} is disjoint with all minimal paths from T_1 to T_{j-1} . The T_i 's are arranged such that the paths with fewer components are numbered first.

To select T_{2D} from T_2 , decompose T_2 into two terms according to a component $k_1, k_1 \in T_1$ and $k_1 \notin T_2$:

$$T_2 = T_2 k_1 \cup T_2 \bar{k}_1. \quad (2.17)$$

If $T_2 k_1 \in T_1$, it is dropped from further consideration because it is already included in T_1 ; otherwise, $T_2 k_1$ is further decomposed according to another component $k_2, k_2 \in T_1$ and $k_2 \notin T_2$, etc. At the same time, if $T_2 \bar{k}_1 \cap T_1 = \Phi$, $T_2 \bar{k}_1$ is disjoint with T_1 and $T_{2D} = T_2 \bar{k}_1$; otherwise, $T_2 \bar{k}_1$ is further decomposed according to another component k_2 etc. The key point is that this procedure decomposes T_j in consideration into two disjoint terms, according to a component k such that $k \in T_i, i < j$, and $k \notin T_j$ and continues the process until all terms disjoint with $T_i, i < j$, are found.

In implementing this algorithm, m_T n -dimensional vectors, $E_i (i = 1, 2, \dots, m_T)$, are defined. E_i corresponds to minimal path T_i such that element k of this vector is 1

if the corresponding component is in the minimal path and 0 otherwise. Also defined is another set of vectors, $P_j \equiv \sum_{i \leq j} E_i$, $j = 1, 2, \dots, m_T$. Therefore, to compare T_j with T_i , $i < j$, it is only necessary to compare P_j with E_j . If there are any non-zero entries in P_j corresponding to zero entries in E_j , the corresponding components are those included in T_i ($i < j$) but not in T_j . Then, branching on each of these components is done. In this way, the algorithm is made more efficient because less comparisons need to be done.

Abraham proposed a theorem which enables the disjoint products to be found much faster [1]. At each stage m , the minimal path T_m is compared successively with T_1, T_2, \dots and T_{m-1} , identifying those components in T_i with values unspecified in T_m , $i < m$. The disjoint term, T_{mD} , results from this series of comparisons. Certain simplifying operations such as dropping terms which are already included are performed.

At stage m of the algorithm, the minimal path T_m is made disjoint with each of the minimal paths T_1, T_2, \dots, T_{m-1} . In the process, T_m is expanded to a set of disjoint products PD_m , each product in this set being disjoint with T_1, T_2, \dots, T_{m-1} . The following theorem is used to do this efficiently.

Theorem: Let T_j be a Boolean product (a minimal path) with only uncomplemented variables and P_i be any of these products.

1. *If there is at least one variable which exists (uncomplemented) in T_j and complemented in P_i , then T_j and P_i are disjoint.*
2. *If T_j and P_i are not disjoint, let $X' \equiv \{x_a, x_b, \dots, x_c\}$ be the set of variables which exists (uncomplemented) in T_j and does not exist in P_i . Then*

- (a) If $X' = \Phi$, then $T_j \cup P_i = T_j$ (the terms in P_i are contained in T_j).
- (b) If $X' \neq \Phi$, then $T_j \cup P_i = T_j \cup \bar{x}_a P_i \cup x_a \bar{x}_b P_i \cup \dots \cup x_a x_b \dots \bar{x}_c P_i$ and all the products in the right hand side are mutually disjoint.

In the above theorem, $\bar{x}_a P_i$, $x_a \bar{x}_b P_i$, ..., and $x_a x_b \dots \bar{x}_c P_i$ are disjoint terms with each other and with T_j . Thus, P_i is made disjoint with T_j by replacing P_i with $(\bar{x}_a P_i \cup x_a \bar{x}_b P_i \cup \dots \cup x_a x_b \dots \bar{x}_c P_i)$.

It was also pointed in Abraham's paper that close approximations to reliability of large networks without excessive computation were possible. When the component unreliabilities are small enough, the contribution of a product term to system reliability decreases rapidly as the number of complemented components in a product term increases. This idea can be used to make modifications to the algorithm to keep track of the number of complemented components in a product term and stop comparing the product when the number of complemented components exceeds a given value. This algorithm compares favorably in computer time with AMG and very favorably with F&M, according to Abraham [1].

Locks reviewed the sum of disjoint products method applied to 2-terminal system reliability problems [53]. He covered three different SDP algorithms that had been published, as discussed above, developed a theory common to all three algorithms, and showed the differences among them.

There are two types of recursive steps in SDP: the outer loop, and a series of inner loops generated by inverting and reinverting components from prior steps. Each inner step results in a term that is disjoint with all the preceding terms. The sum of the term probabilities for the incumbent path is the net increment of system reliability

accounted for by the path. Locks points out that the inner loop procedure is a mirror image of the outer loop [54].

Outer Loop Procedure:

$$\begin{aligned} o(\text{System}) &= T_1 \cup T_2 \cup \dots \cup T_{m_T} \\ &= T_1 + \bar{T}_1 T_2 + \bar{T}_1 \bar{T}_2 T_3 + \dots + \bar{T}_1 \bar{T}_2 \dots \bar{T}_{m_T-1} T_{m_T}. \end{aligned} \quad (2.18)$$

Inner Loop Procedure: Let minimal path T have i fixed 2-valued indicators x_1, \dots, x_i , then

$$\begin{aligned} \bar{T} &= \overline{x_1 x_2 \dots x_i} \\ &= \bar{x}_1 + x_1 \bar{x}_2 + x_1 x_2 \bar{x}_3 + \dots + x_1 x_2 x_3 \dots x_{i-1} \bar{x}_i. \end{aligned} \quad (2.19)$$

For the reliability evaluation of the k -out-of- n systems, the SDP method generates $m_T = \binom{n}{k}$ terms. Each term contributes positively to system reliability. However, it involves many terms in the intermediate steps which disappear in the final expression. For finding a new disjoint product, one has to make it disjoint with all preceding terms. Thus, SDP is at least order of $\binom{n}{k}^2$, or n^{2k} [38].

Improved inclusion-exclusion method

Heidtmann presented an improved IE method specifically for k -out-of- n systems with canceling terms completely eliminated [37]. The method used the following theorem.

Theorem: Consider a k -out-of- n :G system and define the approximation to sys-

tem reliability R_s of step m by

$$R_m = \sum_{j=k}^{m+k-1} (-1)^{j-k} \binom{j-1}{k-1} \sum_{I \subset N_j} \Pr \{ \cap_{i \in I} E_i \} \quad (2.20)$$

$$= R_{m-1} + (-1)^{m-1} \binom{m+k-2}{k-1} \sum_{I \subset N_{m+k-1}} \Pr \{ \cap_{i \in I} E_i \} \quad (2.21)$$

$$R_0 \equiv 0.$$

Then the following bound holds

$$R_{m_e} \leq R_s \leq R_{m_o} \quad m_e, m_o \in N_{n-k+1} \quad (2.22)$$

and

$$R_s = R_{n-k+1}. \quad (2.23)$$

For the case where all the components are s -independent, (2.21) can be written as

$$R_m = R_{m-1} + (-1)^{m-1} \binom{m+k-2}{k-1} \sum_{I \subset N_{m+k-1}} \prod_{i \in I} p_i. \quad (2.24)$$

This method does not use the concept of minimal path. For the reliability calculation, it needs $n - k + 1$ steps. For the unreliability calculation, it needs k steps. We can use $\min\{k, n - k + 1\}$ to select a formula for system reliability evaluations. It is order $\sum_{j=1}^k \binom{n}{j}$, or kn^k , if $k < n - k + 1$. The usual IE method is order $2^{\binom{n}{k}}$. Thus, the improved method is much better than IE for k -out-of- n systems. It is also much better than SDP for k -out-of- n systems, according to Heidtmann [38].

This method reduces the number of terms generated considerably by avoiding cancelling terms. However, each term has to be multiplied by a positive or negative constant which represents the number of repetitions of the term.

Generating function method

In 1984, Barlow and Heidtmann [5] presented two BASIC programs for reliability evaluation of k -out-of- n :G system with independent components. The programs use the following generating function:

$$g_n(z) = \prod_{i=1}^n (q_i + p_i z). \quad (2.25)$$

where z is a dummy variable. Expand and sum coefficients of z^j for $j = k, \dots, n$ to obtain system reliability.

In the form of a BASIC program, the algorithm uses very efficient iterations to compute system reliability of a k -out-of- n system. However, it was not well explained in the paper [5] how the generating function leads to the algorithm. Rushdi [70] explained this algorithm. In fact, the algorithm uses the relation

$$R(k, n) = \sum_{j=k}^n R_e(j, n), \quad (2.26)$$

where $R_e(j, n)$ is the coefficient of z^j in the generating function and it is the probability that exactly j components out of n are working. $R(k, n)$ is the reliability of a k -out-of- n :G system. The program obtains $R_e(j, n)$ through the recursive relation

$$R_e(i, j) = q_j R_e(i, j-1) + p_j R_e(i-1, j-1), \quad (2.27)$$

which is obtained through the construction of

$$g_{j-1}(z) = \prod_{i=1}^{j-1} (q_i + p_i z) = \sum_{i=0}^{j-1} R_e(i, j-1) z^i. \quad (2.28)$$

Hence, the comparison of coefficients in

$$\sum_{i=0}^j R_e(i, j) z^i = (q_j + p_j z) \sum_{i=0}^{j-1} R_e(i, j-1) z^i. \quad (2.29)$$

The solution in (2.27) is achieved with the aid of boundary conditions

$$R_e(-1, j) = R_e(j + 1, j) = 0, \quad \text{for } j = 1, 2, \dots, n. \quad (2.30)$$

The computational complexity of Formula (2.26) is order $n^2/2$. This complexity improves by avoiding unnecessary calculations, and bypassing some other calculations through the use of

$$R(k, k) = R_e(k, k) \quad (2.31)$$

$$R(k, j) = R(k, j - 1) + p_j R_e(k - 1, j - 1), \quad \text{for } k < j. \quad (2.32)$$

The complexity now is order of $k(n - k - 1)$.

Rushdi [70] has proven that this algorithm is the most efficient and is indifferent to which system is selected, i.e., either k -out-of- n :G or $(n - k + 1)$ -out-of- n :F. The reason is that the complexity is $k(n - k - 1)$. Whether k is substituted by $n - k + 1$ does not make any difference.

This algorithm is as efficient as the one presented by Rushdi [70]. However, Barlow and Heidtmann did not present the algorithm well for hand calculations. Because of the bad presentation and lack of explanation of the algorithm, some authors refer to it as a bad algorithm.

JG-1 method

Jain and Gopal [43] proposed an algorithm for computing recursively the exact system reliability of k -out-of- n systems with independent components. It has been referred to as JG-1 method.

Define

$$E(j, i) \equiv \Pr\{\text{exactly } j \text{ components out of } i \text{ are good}\}$$

$$H(i, \Psi) \equiv \Pr\{\text{exactly } i \text{ out of } (\Psi + i) \text{ units are failed}\}$$

$$S(k, k - i) \equiv \text{reliability of } k\text{-out-of-}(k - i)\text{:G system.}$$

Initialize

$$H(0, k - 1) = \prod_{j=1}^{k-1} p_j \quad (2.33)$$

$$S(k, k) = p_k H(0, k - 1). \quad (2.34)$$

At step i of the algorithm.

$$H(i, k - 1) = q_{k-1+i} H(i - 1, k - 1) + p_{k-1-i} H(i, k - 2) \quad (2.35)$$

$$S(k, k + i) = S(k, k - 1 + i) + p_{k+i} H(i, k - 1) \quad (2.36)$$

$$R_s = S(k, n). \quad (2.37)$$

The algorithm finishes when $i = n - k$.

This algorithm generates $\binom{n}{k}$ terms. It, in fact, uses Bayesian Theorem. However, it was not efficiently presented in this paper. In addition, only one of $S(k, k + i)$ and $H(i, \Psi)$ is needed. That both are kept in the algorithm is a waste of efforts and makes the algorithm look more complicated than it really is. It is a worse algorithm than the generating function method even though the author claimed that it was better.

RSPK method

Rai, Sarje, Prasad, and Kumar proposed the following recursive formula for k -out-of- n :G system reliability evaluation [67]. The name RSPK is after the last name initials of the authors.

With definitions,

$$H(s, t) \equiv \Pr\{\text{at least } t \text{ out of } s \text{ units are good}\}, \text{ the units are numbered as } n-s+1, n-s+2, \dots, n$$

$$R(n, k) \equiv \text{reliability of a } k\text{-out-of-}n\text{:G system.}$$

the following formulas are provided:

Let s and t be positive integers such that $s \geq t$, then

$$\begin{aligned} H(s, t) = & p_{n-s+1}H(s-1, t-1) + q_{n-s+1}p_{n-s+2}H(s-2, t-1) \\ & + q_{n-s+1}q_{n-s+2}H(s-2, t). \end{aligned} \quad (2.38)$$

For $s = t$, we have

$$H(s, s) = \prod_{i=n-s+1}^n p_i, \quad \text{for } s \neq 0. \quad (2.39)$$

$$H(s, t) = \begin{cases} 0; & s < t, \\ 1; & t = 0. \end{cases} \quad (2.40)$$

The following compact formula is given for a k -out-of- n :G system:

$$R(n, k) = \sum_{j=1}^k G_j H(n-j, k-j+1) + G_{k+1} H(n-k, 0) \quad (2.41)$$

where.

$$G_j = \begin{cases} \prod_{i=1}^k p_i, & j = k + 1; \\ q_j \prod_{i=1}^{j-1} p_i, & 1 \leq j \leq k. \end{cases} \quad (2.42)$$

With (2.38) to expand $H(s, t)$, (2.41) is used to calculate the reliability of a k -out-of- n :G system.

The number of terms generated by this method is $\binom{n}{k}$. For all k , $1 \leq k \leq n$, computation time increases as $k \rightarrow n/2$ and reaches its maximum for $k = n/2$. For a constant k , computational complexity is bounded by $n^2/4$. For any given n , space complexity of the method is bounded by $k^2/2$. This method uses the same idea as in the JG-1 method, but it is better organized and easier to understand. Because $H(i, \Psi)$ is eliminated, it is a faster algorithm than JG-1. However, there are still some useless computations included in the algorithms. Formula (2.38) actually uses double component decomposition. A simpler method is to use single component decomposition, which will make the algorithm easier to understand and more efficient by eliminating useless terms.

Rushdi's method

Rushdi [70] published a method for computing k -out-of- n system reliability when the components are statistically independent. Initially, the system structure functions are recognized to be monotonically non-decreasing functions. Sequentially, the application of an expansion theorem leads to a recursive relation that governs the required system reliability or unreliability computation. Direct solution of the recursive relation requires only $k(n - k + 1)$ multiplications and yields the numerical value of

k -out-of- n reliability together with a set of meaningful intermediate numerical results that can be useful in the economic assessment of redundancy. A comparison is made of the numerical round-off errors encountered when handling either unreliabilities or reliabilities. Finally the method is compared to some of the existing methods for evaluating k -out-of- n system reliabilities. It is also shown to have a complexity equal to that of the revised program in the Barlow and Heidtmann paper [5]. Hence, it is believed to be optimal.

Define

$S(k, n), \bar{S}(k, n)$ indicator variable for successful and unsuccessful operation of the system, called the system success and system failure, respectively.

$R(k, n), F(k, n)$ reliability and unreliability of the k -out-of- n :G system:

$$R(k, n) = \Pr\{S(k, n) = 1\}$$

$$F(k, n) = \Pr\{\bar{S}(k, n) = 1\} = 1 - R(k, n).$$

The components are assigned fixed serial numbers $(1, 2, \dots, n)$. If n is replaced by j ($1 \leq j \leq n$) in any of the quantities defined above, that quantity describes a system composed of the first j components of the original system (i.e., $1, 2, \dots, j$).

k -out-of- n :G systems are coherent systems. Hence, their successes are monotonically non-decreasing binary functions, and their failures are monotonically non-increasing binary functions [68].

The expansion formula used for $k = 1, 2, \dots, n$ is

$$S(k, n) = \bar{x}_n S(k, n-1) + x_n S(k-1, n-1), \quad (2.43)$$

$$\bar{S}(k, n) = \bar{x}_n \bar{S}(k, n-1) + x_n \bar{S}(k-1, n-1). \quad (2.44)$$

For $k = 0$ or $k = n + 1$, the following relations hold

$$S(0, n) = \bar{S}(n+1, n) = 1, \quad (2.45)$$

$$S(n+1, n) = \bar{S}(0, n) = 0. \quad (2.46)$$

The expansions (2.43) and (2.44) are in sum-of-disjoint-product forms, and hence, are immediately convertible to the following algebraic reliability expressions [69].

$$R(i, j) = q_j R(i, j-1) + p_j R(i-1, j-1) \quad (2.47)$$

$$F(i, j) = q_j F(i, j-1) + p_j F(i-1, j-1) \quad (2.48)$$

Equations (2.47) and (2.48) are recursive relations that are valid for $1 \leq i \leq j$. Each can be solved with the aid of certain boundary conditions which hold for $i = 0$ and $i = j + 1$, and is obtained from (2.45) and (2.46), namely

$$R(0, j) = F(j+1, j) = 1, \quad (2.49)$$

$$R(j+1, j) = F(0, j) = 0. \quad (2.50)$$

Solutions for the reliability $R(k, n)$ or unreliability $F(k, n)$ is easily achieved by programming in languages that allow a program to call itself recursively. However, a closer look at the recursive relations (2.43) and (2.44) reveals that they can be easily represented by a very simple signal flow graph (SFG) structure [47,28]. As an illustration, Figure 2.2 shows the signal flow graph required for the computation of $R(3, 7)$. In that figure, a node at position (i, j) represents $R(i, j)$. The black nodes at $i = 0$ are "source" nodes with unity values, i.e., $R(0, j) = 1$. The black nodes at

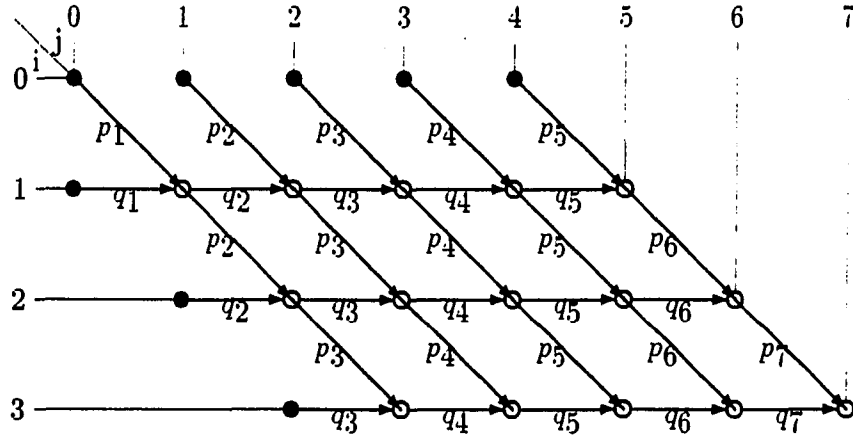
$i = j - 1$ are "source" nodes with zero values, i.e., $R(j + 1, j) = 0$. The same graph in Figure 2.2 can be used for the computation of $F(3,7)$ provided the graph nodes (i, j) are understood to represent the unreliabilities $F(i, j)$ instead of the reliabilities $R(i, j)$, and the two types of source nodes interchange their values, i.e., the black nodes at $i = 0$ become zero values ($F(0, j) = 0$) and the black nodes at $i = j - 1$ become unity values ($F(j + 1, j) = 1$). This algorithm proceeds efficiently by directly constructing (i.e., computing the element values of) the parallelogram array $R(i, j)$ or $F(i, j)$ bounded by $i = 1$, $i = k$, $i = j$, and $i = j - n + k$ (inclusive). The number of elements in that array is $k(n - k + 1)$, and hence, the computational complexity of the algorithm is order $k(n - k + 1)$. In fact, each element of the array requires a single multiplication and two additions for its evaluation. This can be easily seen by invoking the relations ($q_j = 1 - p_j$) to simplify (2.47) and (2.48) into the forms:

$$R(i, j) = R(i, j - 1) + p_j(R(i - 1, j - 1) - R(i, j - 1)), \quad (2.51)$$

$$F(i, j) = F(i - 1, j - 1) + q_j(F(i, j - 1) - F(i - 1, j - 1)). \quad (2.52)$$

It is interesting to note that the present algorithm has the same complexity for its reliability and unreliability versions. This behavior differs from that of most other algorithms; since for those algorithms, a preference exists for one version over the other depending on whether $k \leq n/2$.

Construction of the $R(i, j)$ or $F(i, j)$ array can be processed rowwise, columnwise, or even diagonalwise. However, to minimize the storage requirements, this is done columnwise, for the $R(3, 7)$ case, with due attention paid to the parallelogram boundaries. In this case the algorithm requires storage of $(4 = k + 1)$ scalars only. The storage requirement for any problem is $\min\{k, n - k + 1\}$.

Figure 2.2: Signal flow graph for obtaining $R(3,7)$ and $F(3,7)$ Table 2.1: Reliability array for calculating $R(5,8)$

	0	1	2	3	4	5	6	7	8
0	1.000	1.000	1.000	1.000					
1	.0000	.9000	.9890	.9987	.9983				
2		.0000	.8010	.9664	.9945	.9991			
3			.0000	.7049	.9324	.9858	.9971		
4				.0000	.6132	.8878	.9711	.9929	
5					.0000	.5274	.8337	.9491	.9855

Typical output of the algorithm for the reliability version is shown in Table 2.1. Table 2.1 shows the reliability calculation for $R(5,8)$ when the component reliabilities are $p_j = 0.9 - 0.01(j - 1)$.

An important advantage of the present algorithm is now apparent. All the intermediate results needed for calculating $R(k, n)$ or $F(k, n)$ are meaningful numbers that represent $R(i, j)$ or $F(i, j)$ for $1 \leq i \leq k$ and $i \leq j \leq n - k - i$. These numbers are available to the reliability engineer at no extra cost, and can enable him or

her to make a valid economic assessment of redundancy. For example, row 5 in Table 2.1 represents the reliability $R(5, j)$, where j varies from 5 to 8. The incremental reliability

$$\Delta R(j - 1) = R(5, j + 1) - R(5, j) \quad (2.53)$$

can be calculated easily. The money equivalent of this incremental reliability can therefore be estimated and compared to the cost of adding an additional component, thereby obtaining the optimal number of components for the 5-out-of- j system.

Summary

Reliability evaluation techniques for k -out-of- n systems are reviewed in this chapter. The IE method and the SDP method can be used to compute k -out-of- n system reliabilities, but they are not efficient because they do not utilize the special structure of k -out-of- n systems. The improved IE method eliminates the canceling terms in IE method. The JG-1 method presents a good idea — use Bayesian Theorem to compute k -out-of- n system reliabilities recursively. RSPK uses the same idea as the JG-1.

Rushdi's method is the most efficient and best explained. The generating function method is as efficient as Rushdi's method and a computer program is presented as well.

Much research has been done on the reliability evaluation of k -out-of- n systems. However, more research will be done in this study to further investigate the properties of k -out-of- n systems. Issues such as incremental reliability and component importance of k -out-of- n systems will be covered in Chapter 4.

CHAPTER 3. REVIEW OF CONSECUTIVE- k -OUT-OF- n SYSTEMS

This chapter reviews the research on the consecutive- k -out-of- n systems. First, notation and assumptions are defined. Then, three area of research are covered, (1) reliability evaluation, (2) bounds on system reliability, and (3) system design.

Notation and Assumptions

Notation

n	number of components in a system
k	minimum number of consecutive good (bad) components required for the system to function (fail)
p	component reliability of a system with i.i.d. components
q	component unreliability of system with i.i.d. components; $q = 1 - p$
p_i	reliability of component i in the system, $i = 1, 2, \dots, n$
q_i	unreliability of component i in the system; $q_i = 1 - p_i$, $i = 1, 2, \dots, n$
I_i	reliability importance of component i
$[a]$	the largest integer less than or equal to a
$l(n; k)$	lower bound on reliability of a linear consecutive- k -out-of- n system
$u(n; k)$	upper bound on reliability of the linear system

$l_c(n; k)$	lower bound on reliability of a circular consecutive- k -out-of- n system
$u_c(n; k)$	upper bound on reliability of the circular system
$R(n; k)$	reliability of a linear consecutive- k -out-of- n system
$Q(n; k)$	unreliability of a linear consecutive- k -out-of- n system: $Q(n; k) = 1 - R(n; k)$
$R_c(n; k)$	reliability of a circular consecutive- k -out-of- n system
$Q_c(n; k)$	unreliability of a circular consecutive- k -out-of- n system: $Q_c(n; k) = 1 - R_c(n; k)$
$R(j; k)$	reliability of a linear consecutive- k -out-of- j system: $j = 1, \dots, n$. sometimes it is explicitly denoted by $R(p_1, \dots, p_j; k)$
$Q(j; k)$	unreliability of the linear system: $Q(j; k) = 1 - R(j; k)$
$R_i(n-1; k)$	reliability of a linear consecutive- k -out-of- $(n-1)$ subsystem consisting of components $i+1, \dots, n, 1, \dots, i-1$, for $i = 1, \dots, n$. sometimes it is explicitly denoted by $R_i(p_{i+1}, \dots, p_n, p_1, \dots, p_{i-1}; k)$
$R'(j; k)$	reliability of a linear consecutive- k -out-of- j subsystem consisting of components $n-j+1, \dots, n$. sometimes it is explicitly denoted by $R(p_{n-j+1}, \dots, p_n; k)$
$R_c(j; k)$	reliability of a circular consecutive- k -out-of- j system; $j = 1, \dots, n$
$Q_c(j; k)$	unreliability of the circular system; $Q_c(j; k) = 1 - R_c(j; k)$
$R((i, j); k)$	reliability of a linear consecutive- k -out-of- $(j-i+1)$ subsystem consisting of components $i, i+1, \dots, j$.
$Q((i, j); k)$	unreliability of the linear subsystem; $Q((i, j); k) = 1 - R((i, j); k)$
$Q^*(n; k)$	unreliability of a strict linear consecutive- k -out-of- n :F system

$Q_C^*(n; k)$ unreliability of a strict circular consecutive- k -out-of- n :F system

$R_*(n; k)$ reliability of a relayed linear consecutive- k -out-of- n :F system

Assumptions

- In a circular system of n components, all components are numbered clockwise in increasing order.
- Each component has only two states: good or failed.
- All the components are statistically independent.
- In a consecutive- k -out-of- n :F system:

$$R(j; k) = \begin{cases} 1. & \text{if } 0 \leq j < k, \\ 0. & \text{if } j < 0; \end{cases}$$

In a consecutive- k -out-of- n :G system:

$$R(j; k) = 0, \quad \text{if } j < k.$$

The Consecutive- k -out-of- n :F Systems

System reliability evaluation

Kontoleon reported the first study of the consecutive- k -out-of- n :F system in the literature in 1980 [45]. In the paper, an algorithm was described for obtaining the reliability of a consecutive- k -out-of- n :F system with independent components. The algorithm generates all state combinations of n components with at least k components failed. Then, those combinations with at least k consecutive failures are

identified. The probabilities of the occurrences of these state combinations are added together to obtain the failure probability of the system.

This algorithm generates $\binom{n}{k}$ different terms. Each term is checked to see whether a cut is formed. If a term forms a cut, $n - 1$ multiplications are needed to find the probability of this term. It is an enumeration method, and therefore not efficient.

Chiang and Niu presented the first mathematical formula to compute the exact system reliability of a linear consecutive- k -out-of- n :F system with i.i.d. components [19].

$$R(n; k) = \sum_{r=1}^{n-k+1} \sum_{m=r-1}^{r+k-1} (R(n-m; k) p^r q^{m-r}) + p^{n-k+1}, \quad (3.1)$$

where r denotes the first failed component in the sequence, and m denotes the first functioning component after position r .

This formula is recursive. With proper programming efforts, its complexity is $O(kn)$. Also developed in the paper is a closed formula for the reliability of a consecutive-2-out-of- n :F system.

$$R(n; 2) = \sum_{j=0}^{\lfloor (n+1)/2 \rfloor} \binom{n-j+1}{j} q^j p^{n-j} \quad (3.2)$$

Bollinger and Salvia developed a counting scheme for determining the reliability of a consecutive- k -out-of- n :F system with i.i.d. components in 1982 [15]. The formulas they used are

$$Q(n; k) = \sum_{i=0}^{n-k} r_{i,k,n} p^{n-(k+i)} q^{k+i}, \quad (3.3)$$

$$r_{i,k,n} = \sum_{j=0}^i N(k+i, k+j; n),$$

where $N(k+i, k+j; n)$ is the number of configurations of n components having $(k+i)$ total failures and $(k+j)$ of these consecutive.

The computation of $N(k+i, k+j; n)$ depends on i and j . For those i 's and j 's on the boundaries, some simple combinatorial results are used to generate $N(k+i, k+j; n)$ directly. However, for other i 's and j 's, enumeration must be used. As a result, it is not an efficient algorithm. The paper provided values of $r_{i,k,n}$ for $n \leq 12$.

Derman, Lieberman, and Ross introduced the concept of circular consecutive- k -out-of- n :F system and provided recursive formulas for both the linear and circular consecutive- k -out-of- n :F systems with i.i.d. components [22]. The reliability of the linear consecutive- k -out-of- n :F system is:

$$R(n; k) = \sum_{j=0}^n N(j, n-j+1; k-1) p^{n-j} q^j \quad (3.4)$$

$$N(j, r; m) = \begin{cases} \binom{r}{j}, & \text{if } 0 \leq j \leq r, m = 1; \\ 0, & \text{if } j > r, m = 1; \\ \sum_{i=0}^r \binom{r}{i} N(j-mi, r-i; m-1), & \text{if } m \geq 2, \end{cases} \quad (3.5)$$

where $N(j, r; m)$ is defined as the number of ways in which j identical balls can be placed in r distinct urns subject to the requirement that at most m balls may be placed in one urn. The complexity of this formula is $O(n^2)$.

The recursive formula for circular consecutive- k -out-of- n :F systems with i.i.d. components is:

$$R_c(n; k) = p^2 \sum_{i=0}^{k-1} (i+1) q^i R(n-i-2; k). \quad (3.6)$$

This formula reduces a circular system's reliability evaluation problem into a linear system's reliability evaluation problem. Its complexity depends on the complexity

function of the reliability of a linear consecutive- k -out-of- n :F system. If Formula (3.4) is used for a linear system's reliability evaluation, the complexity of Formula (3.6) is $O(kn^2)$. However, Formula (3.6) is a very simple formula and can be extended to the non-i.i.d. case system reliability evaluation.

Hwang reported the first reliability evaluation for consecutive- k -out-of- n :F systems with independent but not necessarily identical component reliabilities [40].

$$Q(n; k) = Q(n-1; k) + R(n-k-1; k)p_{n-k} \prod_{j=n-k+1}^n q_j, \quad (3.7)$$

where $p_0 \equiv 1$. When the components are i.i.d. the following formula results,

$$Q(n; k) = Q(n-1; k) - pq^k R(n-k-1; k). \quad (3.8)$$

The complexity of the above formulas is $O(n)$.

The circular system is reduced to a sublinear system using the following formula.

$$R_c(n; k) = \sum_{s-1+n-l < k} \left(p_s \prod_{i=1}^{s-1} q_i \right) \left(p_l \prod_{j=l+1}^n q_j \right) R((s+1, l-1); k) \quad (3.9)$$

When the components are i.i.d. the following formula results,

$$R_c(n; k) = \sum_{s-1+n-l < k} p^2 q^{n-l-s-1} R(l-s-1; k). \quad (3.10)$$

Note that s is the first functioning component, while l is the last functioning component in the sequence. The complexity of this formula is $O(nk^2)$.

Shanthikumar reported an algorithm computing the reliability of a linear consecutive- k -out-of- n :F system with independent components [73]. It coincides with the formula for the linear consecutive- k -out-of- n :F systems by Hwang [40].

Griffith and Govindarajula used a Markov Chain approach and developed recursive formulas for the reliability of a linear consecutive- k -out-of- n :F system with i.i.d. components [36]. The approach is different from most that have been discussed, but it cannot be extended to the system with non-i.i.d. components. The formula is as follows.

$$Q(n; k) = \sum_{m=k}^n f_{0k}^m, \quad (3.11)$$

where

$$\begin{aligned} f_{0k}^k &= q^k \\ f_{0k}^m &= pq^k, \text{ for } k+1 \leq m \leq 2k, \\ f_{0k}^{m+1} &= f_{0k}^m - f_{0k}^{k+1} f_{0k}^{m-k}, \text{ for } m \geq 2k. \end{aligned}$$

Lambiris and Papastavridis derived closed reliability formulas for the linear and circular consecutive- k -out-of- n :F systems with i.i.d. components in 1985 [51]. The formulas look too complicated.

$$R(n; k) = \sum_{i=0}^n \binom{n-ik}{i} (-1)^i (pq^k)^i - q^k \sum_{i=0}^n \binom{n-ik-k}{i} (-1)^i (pq^k)^i, \quad \forall n \quad (3.12)$$

$$\begin{aligned} R_c(n; k) &= k \sum_{i=0}^n \binom{n-ik-k-1}{i} (-1)^i (pq^k)^{i-1} \\ &\quad + \sum_{i=0}^n \binom{n-ik}{i} (-1)^i (pq^k)^i - q^n, \quad n \geq k, \end{aligned} \quad (3.13)$$

where

$$R_c(n; k) = 0, \text{ for } n < 0.$$

Antonopoulou and Papastavridis developed a recursive formula for the circular consecutive- k -out-of- n :F system with independent components in 1987 [4]. It is better

than Hwang's [40] because it has a complexity of $O(kn)$.

$$\begin{aligned}
 R_C(n; k) &= p_n R(n-1; k) + q_n R_C(n-1; k) \\
 &\quad - \sum_{i=0}^{k-1} \left(p_{i+1} \prod_{j=1}^i q_j \right) \left(p_{n-k+i} \prod_{j=n-k+i-1}^n q_j \right) \\
 &\quad \times R((i+2, n-k+i-1); k). \tag{3.14}
 \end{aligned}$$

In summary, the best reliability evaluation formulas for the linear and the circular consecutive- k -out-of- n :F systems with independent components are so far given by Hwang [40] and Antonopoulou and Papastavridis [4], respectively. They have complexities of $O(n)$ and $O(nk)$, respectively, and are listed below.

$$\begin{aligned}
 Q(n; k) &= Q(n-1; k) + R(n-k-1; k) p_{n-k} \prod_{j=n-k+1}^n q_j \\
 R_C(n; k) &= p_n R(n-1; k) + q_n R_C(n-1; k) \\
 &\quad - \sum_{i=0}^{k-1} \left(p_{i+1} \prod_{j=1}^i q_j \right) \left(p_{n-k+i} \prod_{j=n-k+i-1}^n q_j \right) \\
 &\quad \times R((i+2, n-k+i-1); k).
 \end{aligned}$$

Bounds and approximations of system reliability

In many applications, exact system reliability is not needed. Good bounds which can be easily computed are usually sufficient. This section reviews the developments in the bounds for system reliability.

Chiang and Niu presented the first bounds for the consecutive- k -out-of- n :F systems with i.i.d. components [19]. Since the failure of any k consecutive components causes the failure of a consecutive- k -out-of- n :F system, any k consecutive components constitute a minimal cut set. Furthermore, k consecutive components are the only

type of minimal cut set of a consecutive- k -out-of- n :F system and there are $n - k - 1$ minimal cut sets in a linear consecutive- k -out-of- n :F system. If the system is functioning, there is at least one functioning component in every cut set. With this argument, the lower bound for a linear consecutive- k -out-of- n :F system is developed and given below.

$$l(n; k) = (1 - q^k)^{n-k+1} \quad (3.15)$$

To obtain an upper bound, Chiang and Niu partitioned the consecutive- k -out-of- n :F system into $[n/k] + 1$ independent subsystems, where each subsystem had k consecutive components except the last one which had $n - k[n/k]$ components [18]. Since $n - k[n/k] < k$, the last subsystem could not fail. Because the system works, all $[n/k] - 1$ subsystems must work. Hence, the upper bound for the consecutive- k -out-of- n :F system is

$$u(n; k) = (1 - q^k)^{[n/k]}. \quad (3.16)$$

Derman, Lieberman, and Ross developed the lower and upper bounds for the linear and circular consecutive- k -out-of- n :F system with independent components [22]. Similar arguments to that in Chiang and Niu [18] are used for the lower bound development.

$$l(n; k) = \prod_{i=1}^{n-k+1} (1 - \prod_{j=i}^{i+k-1} q_j) \quad (3.17)$$

$$l_c(n; k) = \prod_{i=1}^n (1 - \prod_{j=i}^{i+k-1} q_j), \quad (3.18)$$

where $q_j = q_{j-n}$ for $j > n$. These lower bounds may be improved by calculating the

exact reliability of a subsystem.

$$l(n; k) = \Pr[E_1 E_2 \cdots E_i] \prod_{j=i+1}^{n-k+1} \left(1 - \prod_{m=j}^{j+k-1} q_m\right) \quad (3.19)$$

$$l_c(n; k) = \Pr[E_1 E_2 \cdots E_i] \prod_{j=i+1}^n \left(1 - \prod_{m=j}^{j+k-1} q_m\right), \quad (3.20)$$

where E_i ($1 \leq i \leq n - k$ for the linear case and $1 \leq i \leq n - 1$ for the circular case) is the event that not all the k components, $i, i + 1, \dots, i + k - 1$, fail. The higher the i , the better the lower bound, and the higher the complexity.

The upper bounds for the linear and circular systems are presented in the following formula.

$$u(n; k) = 1 - \frac{E^2[N]}{E[N^2]}, \quad (3.21)$$

where N is a random variable representing the number of minimal cut sets whose components all fail.

$$E[N] = \begin{cases} \sum_{i=1}^{n-k+1} \prod_{j=i}^{i+k-1} q_j, & \text{for the linear case;} \\ \sum_{i=1}^n \prod_{j=i}^{i+k-1} q_i, & \text{for the circular case;} \end{cases} \quad (3.22)$$

with $q_j = q_{j-n}$ if $j > n$.

$$E[N^2] = E\left[\sum_i I_i + \sum_{i \neq j} I_i I_j\right] = \sum_i E[I_i] + \sum_{i \neq j} E[I_i I_j], \quad (3.23)$$

where

$$I_i = \begin{cases} 1, & \text{if all the components in } E_i \text{ are failed;} \\ 0, & \text{otherwise.} \end{cases}$$

In the circular system where all components are i.i.d., the following upper bound is explicitly presented.

$$u_c(n; k) = 1 - \frac{A}{B}, \quad (3.24)$$

where

$$\begin{aligned} A &= nq^k \\ B &= 1 - (n - 2k + 1)q^k + \frac{2q(1 - q^{k-1})}{p}. \end{aligned}$$

Salvia presented formulas of lower and upper bounds for the reliability of a linear consecutive- k -out-of- n :F system with i.i.d. components [71].

$$l(n; k) = 1 - (n - k + 1)q^k \quad (3.25)$$

$$u(n; k) = 1 - (n - k + 1)p^{n-k}q^k. \quad (3.26)$$

Griffith and Govindarajulu provided the following approximations to the reliability of a consecutive- k -out-of- n :F system with i.i.d. components for large n [36].

$$R(n; k) \approx \frac{1 - qx}{(k + 1 - kx)px^{n+1}}, \quad (3.27)$$

where x is the unique positive root of s different from $1/q$ in the following equation.

$$pq^k s^{k+1} - s + 1 = 0$$

The accuracy of the above approximation is ensured to be within 5 decimal places for n in the range of 10 to 15 for given p and k as reported in [36].

Fu developed the upper and lower bounds for the reliability of a large consecutive- k -out-of- n :F system with i.i.d. components [30]. For every small $\delta > 0$, there exists $n_0(\delta)$ such that for all $n \geq n_0(\delta)$, the lower and upper bounds for the reliability of a consecutive- k -out-of- n :F system are:

$$l(n; k) = \left(1 - \frac{\lambda^k}{n}\right)^{n-k+1} \quad (3.28)$$

$$\begin{aligned}
u(n; k) &= \left(1 - \frac{\lambda^k}{n} - \frac{(1+\delta)\lambda^{k+1}}{n^{1+1/k}} \right)^{n-n_0(\delta)} \\
n_0(\delta) &= \begin{cases} k, & \text{if } n^*(\delta) < k; \\ n^*(\delta), & \text{if } k \leq n^*(\delta) \leq n; \\ n, & \text{if } n^*(\delta) > n; \end{cases} \\
n^*(\delta) &\equiv \text{integer part of } \left(\frac{(1+\delta)\lambda}{\delta} \right)^k \\
\lambda &= qn^{1/k}.
\end{aligned} \tag{3.29}$$

Fu also examined upper and lower bounds for the reliability of a large consecutive- k -out-of- n :F system with unequal component reliability [31]. The reliability of a large consecutive- k -out-of- n :F system is derived, under certain conditions, from the upper and lower bounds.

$$l(n; k) = \prod_{i=k}^n (1 - a_{i:k,n}) \tag{3.30}$$

$$u(n; k) = \prod_{i=k}^n (1 - b_{i:k,n}), \tag{3.31}$$

where

$$\begin{aligned}
a_{i:k,n} &= \prod_{j=i-k+1}^i q_j \\
b_{i:k,n} &= \prod_{j=i-k+1}^i q_j - \frac{q_{i-k}}{p_{i-k}} \prod_{j=i-k+1}^i q_j.
\end{aligned}$$

When $n \rightarrow \infty$, the following inequalities exist.

$$e^{-\bar{\theta}} \leq R(k) \leq e^{-\underline{\theta}}, \tag{3.32}$$

where

$$R(k) = \lim_{n \rightarrow \infty} R(k; n)$$

$$\bar{\theta} = \lim_{n \rightarrow \infty} \sum_{j=1}^n \left(\frac{1}{j} \sum_{i=k}^n a_{i:k,n}^j \right)$$

$$\underline{\theta} = \lim_{n \rightarrow \infty} \sum_{j=1}^n \left(\frac{1}{j} \sum_{i=k}^n b_{i:k,n}^j \right).$$

Results in Chao and Lin [16] and Fu [30] are special cases of the above formulas. Papastavridis derived the upper and lower bounds for linear and circular consecutive- k -out-of- n :F systems with i.i.d. components when the component reliability is less than $k/(k-1)$ [62]. He found the formulas by analyzing the roots of the denominator of the generating functions of the reliability of a linear and circular consecutive- k -out-of- n :F systems.

$$l(n; k) = bm^{n+1} - e \quad (3.33)$$

$$u(n; k) = aM^{n+1} + e \quad (3.34)$$

$$l_c(n; k) = M^n - (k-1)q^n \quad (3.35)$$

$$u_c(n; k) = M^n + (k-1)q^n, \quad (3.36)$$

where

$$b = \frac{M^k - q^k}{M^k - (k+1)pq^k}$$

$$m = 1 - \frac{pq^k}{(1 - q^k)^k}$$

$$e = \frac{2(k-1)q^{n+2}}{p(k + (k+1)q)}$$

$$M = 1 - pq^k$$

$$a = \frac{m^k - q^k}{m^k - (k+1)pq^k}.$$

Optimal system design

In a system with many components, we know that all components contribute to the system's performance if the system is coherent. However, which component plays a more important role to the system's function depends on the system's structure. As a result, the system's design is critical to the system's performance given the resources available.

The optimal design problem of a consecutive- k -out-of- n system was first studied by Derman et al. [22], which can be briefly described as the following. Let p_1, \dots, p_n denote the component reliabilities. Suppose that the reliabilities of the n components available have been ordered and let

$$p_1 \leq p_2 \leq \dots \leq p_n$$

denote their ordered values. The problem of interest concerns the optimal arrangement (permutation) of the components so that the system reliability is maximized.

For the linear consecutive-2-out-of- n :F system, the optimal design was conjectured by Derman et al. [22] and proven by Wei et al. (partially) [78], Malon [56], and Du and Hwang [24], and is given below,

$$(1, n, 3, n-2, \dots, n-3, 4, n-1, 2),$$

i.e., position 1 and n are assigned the least reliable components; the two most reliable components are placed at the two adjacent positions, with the most reliable component paired with the least reliable component and the second most reliable component paired with the second least reliable component, etc.

Hwang [40] extended the conjecture of Derman et al. to the circular consecutive-2-out-of- n :F system by conjecturing that the optimal design of a circular consecutive-2-out-of- n :F system is

$$\tilde{\psi} = (n, 1, n-1, 3, n-3, \dots, n-4, 4, n-2, 2, n),$$

which was also proven by Malon [56] and Du and Hwang [24] independently.

Tong [77] discovered that the system's reliability does not depend on the permutation of the components (or their reliabilities) for a circular consecutive- k -out-of- n :F system when $k = n-1$ or $k = n$. Thus, any permutation of the components is an optimal design.

Malon [57] studied the optimal design of a consecutive- k -out-of- n :F system for all possible k values and discovered that the consecutive- k -out-of- n :F system admits an invariant optimal configuration if and only if $k \in \{1, 2, n-2, n-1, n\}$. He also pointed out that the necessary condition for the optimal design of a consecutive- k -out-of- $2k$:F system is to arrange the leftmost k components in an order of increasing component reliability and the k rightmost components in an order of decreasing component reliability.

Using the reliability importance definition given by Birnbaum [9], Papastavridis [66] provided the following formula for component reliability importance function for a linear consecutive- k -out-of- n :F system,

$$I_i = \frac{(R(i-1; k)R'(n-i; k) - R(n; k))}{q_i}. \quad (3.37)$$

If all components in the system are equally reliable, the author claimed that the most important components are in the middle of the sequence of components. Kuo and Zhang [48] pointed out that the statement by Papastavridis [66] was false.

Kuo and Zhang [48] discussed the component reliability importance of a consecutive- k -out-of- n :F system in details. For a linear consecutive- k -out-of- n :F system with i.i.d. components, all the components are equally important when $k = 1$ and $k = n$ because the system becomes a series and a parallel system respectively. When $k > n/2$, the component importance increases from the first component to the $(n - k - 1)$ st component and decreases from the k th component to the n th component. However, the components between the $(n - k + 1)$ st and the k th have identical importance. When $k \leq n/2$, the component importance increases from position 1 to position k and decreases from position $n - k + 1$ to position n . The importance between k and $n - k - 1$ fluctuates without a fixed pattern.

Based on their interpretation of component importance, Kuo and Zhang [48] developed the following optimal strategy for the design of consecutive- k -out-of- n :F system: (1) given a desired system reliability with to-be-determined component reliabilities, we would devote the minimal effort to allocate higher reliability to more important positions and lower reliability to less important positions; (2) given the reliabilities of n components, the sequence of assigning components to the system should follow the pattern of component importance by allocating high reliability components to high component importance positions; and (3) the above allocations may not be optimal if different costs are incurred when allocating components at different positions.

They also discovered that the necessary condition for the optimal design of a general consecutive- k -out-of- n :F system is to arrange components from position 1 to position $\min(k, n - k + 1)$ in non-decreasing order of component reliability, to arrange

the components from position $\max(k, n - k + 1)$ to position n in non-increasing order of component reliability, and to arrange the $2k - n$ most reliable components in the middle in any order if $n < 2k$.

Another result in their paper is the necessary condition for the optimal design of a circular consecutive- k -out-of- $(k + 2)$:F system, which is

$$(p_i - p_{i+3})(p_{i+1} - p_{i+2}) \leq 0 \quad \text{for } i = 1, 2, \dots, k + 2,$$

where p_i is the reliability of the component at position i and $p_j = p_{j-k+2}$ if $j > k+2$.

This is a very weak condition.

The necessary conditions for the optimal design of consecutive- k -out-of- n :F systems are summarized in Table 3.1. The sufficient conditions are summarized in Table 3.2.

Table 3.1: Necessary conditions of the optimal design of linear and circular consecutive- k -out-of- n :F systems

System	Necessary Condition
Linear	Arrange the leftmost $\min(k, n - k + 1)$ components in non-decreasing order of component reliability and the rightmost $\min(k, n - k + 1)$ components in non-increasing order of component reliability. If $n < 2k$, the $2k - n$ best components should be placed in the middle in any order.
Circular $n = k + 2$	$(p_i - p_{i+3})(p_{i+1} - p_{i+2}) \geq 0$ for $i = 1, 2, \dots, k + 2$

Table 3.2: Summary of invariant optimal designs of linear consecutive- k -out-of- n :F systems

System	Invariant Optimal Designs
$k = 1$	(any configuration)
$k = 2$	$(1, n, 3, n - 2, \dots, n - 3, 4, n - 1, 2)$
$2 < k < n - 2$	(Does not exist)
$k = n - 2$	$(1, 4, (\text{any arrangement}), 3, 2)$
$k = n - 1$	$(1, (\text{any arrangement}), 2)$
$k = n$	(any arrangement)

The Consecutive- k -out-of- n :G Systems

Reliability and reliability bound evaluation

The concept of consecutive- k -out-of- n :G system was brought up in Kuo, Zhang, and Zuo [49]. The relationship between the consecutive- k -out-of- n :F system and the consecutive- k -out-of- n :G system was built up and similar results on system reliability evaluation, reliability bound evaluation, and system design for consecutive- k -out-of- n :G systems were reported.

A consecutive- k -out-of- n :G system consists of an ordered sequence of n components such that the system works if and only if at least k consecutive components in the system are good. The relationship between the consecutive- k -out-of- n :F system and the consecutive- k -out-of- n :G system is described below.

If the reliability of component i , p_i , in one type of consecutive- k -out-of- n system (say, F system) is equal to the unreliability of component i , q_i , in the other type of consecutive- k -out-of- n system (G system) for

$i = 1, 2, \dots, n$, given that both types of systems have the same values of n and k , then the reliability of one type of system is equal to the unreliability of the other type of system.

Due to the relationship between the consecutive- k -out-of- n :F system and consecutive- k -out-of- n :G system described above, and the available results of reliability and reliability bound evaluation for the consecutive- k -out-of- n :F systems, the following formulas were developed easily in Kuo, Zhang, and Zuo [49].

$$R(n; k) = R(n-1; k) + Q(n-k-1; k)q_{n-k} \left(\prod_{i=n-k+1}^n p_i \right) \quad (3.38)$$

$$\begin{aligned} R_c(n; k) &= q_n R(p_1, \dots, p_{n-1}; k) - p_n R_c(p_1, \dots, p_{n-1}; k) \\ &\quad + \sum_{i=1}^k (q_{n-k+i-1} p_{n-k+i} \cdots p_n p_1 \cdots p_{i-1} q_i) \\ &\quad \times Q((i+1, n-k+i-2); k). \end{aligned} \quad (3.39)$$

If the components are i.i.d., we have

$$R(n; k) = R(n-1; k) + Q(n-k-1; k)qp^k \quad (3.40)$$

$$R_c(n; k) = qR(n-1; k) + pR_c(n-1; k) + kq^2p^kQ(n-k-2; k). \quad (3.41)$$

The bounds of the reliability of linear and circular consecutive- k -out-of- n :G systems are

$$l(n; k) = l_c(n; k) = 1 - \prod_{j=0}^{\lceil n/k \rceil - 1} \left(1 - \prod_{i=jk+1}^{jk+k} p_i \right) \quad (3.42)$$

$$u(n; k) = 1 - \prod_{j=1}^{n-k-1} \left(1 - \prod_{i=0}^{k-1} p_{j+i} \right) \quad (3.43)$$

$$u_c(n; k) = 1 - \prod_{j=1}^n \left(1 - \prod_{i=0}^{k-1} p_{j+i} \right). \quad (3.44)$$

When the components in the system are i.i.d., we have the following bounds for the system reliabilities of both linear and circular consecutive- k -out-of- n :G systems.

$$l(n; k) = l_c(n; k) = 1 - (1 - p^k)^{\lceil n/k \rceil} \quad (3.45)$$

$$u(n; k) = 1 - (1 - p^k)^{n-k+1} \quad (3.46)$$

$$u_c(n; k) = 1 - (1 - p^k)^n \quad (3.47)$$

Optimal system design

Using Birnbaum's definition of component reliability importance and the results of consecutive- k -out-of- n :F systems, Kuo, Zhang, and Zuo [49] developed component importance formulas and optimal designs for consecutive- k -out-of- n :G systems.

1. Component Importances:

$$I_i = \frac{1}{p_i} [R(n; k) - R(i-1; k) - R'(n-i; k) + R(i-1; k)R'(n-i; k)]$$

(Linear system) (3.48)

$$I_i = \frac{1}{p_i} [R_c(n; k) - R_i(n-1; k)] \quad \text{(Circular system)} \quad (3.49)$$

In a linear consecutive- k -out-of- n :G system with i.i.d. components, the component importance increases from position 1 to position $\min(k, n-k+1)$ and decreases from position $\max(k, n-k+1)$ to position n . If $n < 2k$, the component importance stays constant between component $n-k+1$ and component k . In a circular consecutive- k -out-of- n :G system with i.i.d. components, all components have equal component importance.

2. The necessary condition for the optimal configuration of a linear consecutive- k -out-of- n :G system is to

- (a) Arrange the components from position 1 to position $\min(k, n - k + 1)$ in a non-decreasing order of component reliability.
 - (b) Arrange the components from position $\max(k, n - k + 1)$ to position n in a non-increasing order of component reliability.
 - (c) Arrange the $2k - n$ best components between $n - k + 1$ and k in any order if $n < 2k$.
3. The optimal configuration of a linear consecutive- k -out-of- n :G system with $n \leq 2k$ is

$$(1, 3, 5, \dots, 2(n - k) - 1, (\text{any arrangement}), 2(n - k), \dots, 6, 4, 2)$$

given that $p_1 < p_2 < \dots < p_{n-1} < p_n$.

4. All arrangements of n components in a circular consecutive- k -out-of- $(k + 1)$:G system give the same system reliability.
5. The necessary condition for the optimal configuration of a circular consecutive- k -out-of- $(k + 2)$:G system is

$$(p_i - p_{i+3})(p_{i+1} - p_{i+2}) \geq 0, \quad \text{for } i = 1, 2, \dots, k + 2,$$

where p_i represents the reliability of the component at position i and $p_j = p_{j-k-2}$ if $k > k + 2$.

Special Consecutive- k -out-of- n :F Systems

Bollinger [13] presented a special version of consecutive- k -out-of- n :F systems. He defines strict consecutive- k -out-of- n :F systems which operate in such a way that

isolated strings of failures of length less than k either do not occur or are immediately corrected. Thus, system failure occurs if and only if k or more consecutive components fail, and without any isolated failure strings of fewer than k consecutive components.

Previous methods of calculating the failure probability for consecutive- k -out-of- $n:F$ systems do not rule out the situation that the failure mode (at least one string of k or more consecutive failures) is also accompanied by any possible strings of isolated failures of length less than k . For example, in a consecutive-3-out-of-10:F system (with F (O) representing a failed (operating) component) the state FOFOFFOFFF is one of system failures only because of the last three Fs, but there are also three isolated failure strings of length less than three. It seems reasonable to suppose, however, that in at least some applications of these systems, as might be the case with communication relay systems, isolated failure strings of length less than k — which may degrade performance but do not cause system failure — are, or can be, detected and corrected within an interval short enough that the normal operating mode can be considered to have no failed components. That is, it is assumed here that although prevention of loss of system continuity is important enough that a consecutive- k -out-of- $n:F$ system design is used for protection, the detection and repair or replacement of isolated failed components occur quickly enough that the context is not of the ordinary consecutive- k -out-of- $n:F$ system. In such a case, system failure will occur when and only when k or more consecutive components fail, *and* without any isolated failure strings of fewer than k consecutive components. Such a system is called a *strict* consecutive- k -out-of- $n:F$ system.

As might be expected, the failure probability for an ordinary system is extremely

conservative compared to that for a strict system. When the strict system applies, it might be possible to use the information this provides in design economies.

A method was given in Bollinger [13] for calculating the failure probability function for strict consecutive- k -out-of- n :F systems with i.i.d. components. It is through the enumeration of the number of binary strings of length n containing a given number of 0's, and in which the 0's occur only in blocks of length at least k .

$$Q^*(n; k) = \sum_{m=k}^n \alpha_{m,n}^{(k)} p^{n-m} q^m, \quad (3.50)$$

where $\alpha_{m,n}^{(k)}$ denotes the number of non-zero binary strings of length n containing m 0's, $k \leq m \leq n$, and in which 0's occur only in blocks of length at least $k \geq 2$.

The problem of obtaining $\alpha_{m,n}^{(k)}$ is solved by an algorithm of array manipulations. The algorithm is easy to program, gives exact calculations for n and k as large as desired, and produces the α 's for all k for a fixed n . An example of a tabulation of the α 's for $n = 9$ was provided in the paper.

Kossow and Preuss [46] extended Bollinger's definition of strict linear consecutive- k -out-of- n :F system to circular consecutive- k -out-of- n :F systems. They presented failure probability function of a strict linear consecutive- k -out-of- n :F system in a closed form. The calculation of a strict circular consecutive- k -out-of- n :F system is reduced to the linear case.

$$Q^*(n; k) = \sum_{m=k}^n \alpha_{m,n}^{(k)} p^{n-m} q^m, \quad (3.51)$$

where

$$\alpha_{m,n}^{(k)} = \sum_{j=1}^{\lfloor m/k \rfloor} \binom{n-m+1}{j} \binom{m-j(k-1)-1}{j-1}. \quad (3.52)$$

For the strict circular consecutive- k -out-of- n :F systems, the following formula reduces its reliability evaluation to a strict linear system reliability evaluation.

$$Q_c^*(n; k) = \sum_{m=k}^{n-1} mp^{n-m}q^m + \sum_{m=k}^{n-k-2} mp^2q^m Q^*(n-m-2; k) + pQ^*(n-1; k) + q^n \quad (3.53)$$

with assumptions: (1) an empty sum is zero and (2) $Q^*(i; k) = 0$, for $i < k$.

Hwang [41] presented another variation of the consecutive- k -out-of- n :F system — relayed consecutive- k -out-of- n :F system. The often quoted examples for consecutive- k -out-of- n :F systems are telecommunications systems, oil pipeline systems, and mobile communication systems [18,19]. In all these examples, some object, be it a message or the flow of a signal, is relayed from a source to a sink through a sequence of intermediate stations. Care should be taken as to whether the source and the sink are also considered components of the systems, i.e., whether they serve the same function as the intermediate stations. In the telecommunications system example and the oil pipeline system example, source, sink, and the intermediate stations are the same type of relay stations in one case and pumping stations in the other. Thus, both source and sink are considered components of the systems. In the mobile communication system example, it is assumed that the source and intermediate stations are all photo-transmitting spacecraft, but not the sink (which could be just an antenna). Thus, the system includes the source as a component but excludes the sink (this is true in the oil pipeline system if the sink is just a storage unit).

In these examples where the source (sink) is also a component of a system, the consecutive- k -out-of- n :F system model does not describe the system accurately, since the system works only if the source (sink) works, regardless of the value of k . Such a

system is named a *relayed consecutive- k -out-of- n :F system*, *unipolar* if only the source is included and *bipolar* if both source and sink are part of the system.

For such relayed consecutive- k -out-of- n :F systems, Hwang presented results on system reliability and optimal system design as follows.

$$R_*(n; k) = \begin{cases} p_1 R(p_2, p_3, \dots, p_n; k), & \text{for the unipolar case;} \\ p_1 p_n R(p_2, p_3, \dots, p_{n-1}; k), & \text{for the bipolar case.} \end{cases} \quad (3.54)$$

Since there exist $O(n)$ time algorithms to compute $R(n; k)$, $R_*(n; k)$ can also be computed in $O(n)$ time.

An invariant optimal design of a unipolar relayed consecutive- k -out-of- n :F system is to assign the most reliable component to position 1 and arrange the remaining components following the invariant optimal permutation for linear consecutive- k -out-of- $(n - 1)$:F system. An invariant optimal design of a bipolar relayed consecutive- k -out-of- n :F system is to assign the two most reliable components to positions 1 and n and arrange the remaining $n - 2$ components following the invariant optimal permutation for linear consecutive- k -out-of- $(n - 2)$:F system.

As the author pointed out, many systems treated as consecutive- k -out-of- n :F systems in the literature are actually different kinds of animals which are called relayed consecutive- k -out-of- n :F systems. Although the reliability formulas for the two models are similar, the actual reliabilities computed are different. Therefore, one should choose the correct model for reliability analysis.

Summary

The most efficient formulas for reliability evaluation of consecutive- k -out-of- n systems developed so far are recursive. Only a few invariant optimal designs for

consecutive- k -out-of- n systems have been reported, especially for circular consecutive- k -out-of- n systems. For some k and n combinations, there may not exist any invariant optimal configuration. Also some methods for k -out-of- n systems may be adopted for consecutive- k -out-of- n systems. More research results on these systems will be reported in the following two chapters.

CHAPTER 4. ANALYSIS OF k -OUT-OF- n AND CONSECUTIVE- k -OUT-OF- n SYSTEMS

k -out-of- n Systems

A k -out-of- n :G system is good if and only if at least k of its n components are good. In a k -out-of- n :G system with i.i.d. components, the probability that exactly k out of n components work is

$$\Pr[\text{Exactly } k \text{ components out of } n \text{ work}] = \binom{n}{k} p^k q^{n-k}. \quad (4.1)$$

Then, we have the following lemma for reliability evaluation of a k -out-of- n :G system with i.i.d. components.

Lemma 1 *A closed formula for reliability evaluation of a k -out-of- n :G system with i.i.d. components is*

$$R(n; k) = p^k \sum_{j=k}^n \binom{j-1}{k-1} q^{j-k}, \quad \text{for } 1 \leq k \leq n. \quad (4.2)$$

Proof of Lemma 1

According to the formulas by Rushdi [70], the system reliability of a k -out-of- n :G system with i.i.d. components is

$$R(n; k) = p[R(n-1; k-1) - R(n-1; k)] + R(n-1; k)$$

$$\begin{aligned}
&= p \Pr[\text{Exactly } k-1 \text{ components out of } n-1 \text{ work}] - R(n-1; k) \\
&= \binom{n-1}{k-1} p^k q^{n-k} + R(n-1; k).
\end{aligned}$$

Now, we have obtained a recursive formula

$$R(j; k) = \binom{j-1}{k-1} p^k q^{j-k} + R(j-1; k), \quad j \geq k, \quad (4.3)$$

or

$$R(j; k) - R(j-1; k) = \binom{j-1}{k-1} p^k q^{j-k}, \quad j \geq k. \quad (4.4)$$

and

$$R(j; k) = 0, \quad \text{for } j < k.$$

As a result,

$$\begin{aligned}
R(n; k) &= \sum_{j=k}^n [R(j; k) - R(j-1; k)] \\
&= \sum_{j=k}^n \binom{j-1}{k-1} p^k q^{j-k} \\
&= p^k \sum_{j=k}^n \binom{j-1}{k-1} q^{j-k}.
\end{aligned}$$

(Q.E.D.)

Lemma 2 For a k -out-of- n : G system with i.i.d. components, define

$$f(n, k, p) = \frac{R(n; k)(k-1)!}{p^k}, \quad (4.5)$$

then,

$$f(n, k+1, p) = \frac{\partial f(n, k, p)}{\partial q} \quad (4.6)$$

Proof of Lemma 2

$$\begin{aligned}
f(n, k, p) &= \frac{R(n; k)(k-1)!}{p^k} = \frac{p^k \sum_{j=k}^n \binom{j-1}{k-1} q^{j-k} (k-1)!}{p^k} \\
&= \sum_{j=k}^n \frac{(j-1)!}{(j-k)!} q^{j-k} \\
\frac{\partial f(n, k, p)}{\partial q} &= \sum_{j=k}^n \frac{(j-1)!}{(j-k)!} (j-k) q^{j-k-1} \\
&= \sum_{j=k+1}^n \frac{(j-1)!}{(j-k-1)!} q^{j-k-1} \\
&= f(n, k+1, p).
\end{aligned}$$

(Q.E.D.)

This result may be useful if there is a method to evaluate a function's derivative efficiently.

The reliability of a k -out-of- n :G system with i.i.d. components is a function of n , k , and p . The increase of n , p , both, or a decrease of k will increase the system's reliability. However, there are component importance factors and cost factors to be considered in improving system reliability.

To analyze the effects of a component's reliability to system reliability, Birnbaum [9] defined component reliability importance. The component reliability importance function of a system is

$$\begin{aligned}
I_i &= \frac{\partial R(n; k)}{\partial p_i} = R(p_1, \dots, p_{i-1}, 1, p_{i+1}, \dots, p_n; k) \\
&\quad - R(p_1, \dots, p_{i-1}, 0, p_{i+1}, \dots, p_n; k). \quad (4.7)
\end{aligned}$$

In a k -out-of- n :G system with i.i.d. components, the reliability importance of the components are all the same.

$$I_i = \binom{n-1}{k-1} p^{k-1} q^{n-k}. \quad (4.8)$$

This formula says that the importance of any single component is equal to the probability that there are exactly $k-1$ components working out of an $n-1$ component system.

When the components of a k -out-of- n :G system are not i.i.d., the reliability importance of component i is

$$I_i = \Pr[\text{Exactly } k-1 \text{ components out of the rest } n-1 \text{ work}]. \quad (4.9)$$

We may think of a k -out-of- n system with k constant and n variable. The importance of adding the $(n+1)$ th component is the probability that there are exactly $k-1$ components working in the original n component system. If in the n component system, $k-1$ components have relatively high reliabilities while the other $n-k+1$ components have relatively low reliabilities, then the $(n+1)$ th component in consideration is relatively more important than otherwise. If there are already k or more than k very good components in the n component system, the adding of an $(n+1)$ th component is not that important. The importance of a component is the probability that adding this component will form the first minimal path for the system.

For example, consider a 1-out-of-2:G system (a parallel system).

$$R(p_1, p_2) = p_1 + p_2 - p_1 p_2$$

$$I_1 = 1 - p_2$$

$$I_2 = 1 - p_1.$$

We can see that the importance of component 1 is the probability that component 2 fails and the importance of component 2 is the probability that component 1 fails. The more reliable component 2, the less important component 1, and vice versa.

Adding one more component to a k -out-of- n :G system will always increase the system's reliability. However, the increase in the system's reliability becomes smaller as n becomes larger, while the cost of adding one more component is the same. If all components available are i.i.d., the optimal value of n is determined when there is no significant improvement on system reliability by adding one more component. Figure 4.1 shows the relationships of system reliability and cost as functions of n , with p and k fixed.

Now look at the effect of component reliability on the reliability of a k -out-of- n :G system with i.i.d. components. If $k = 1$ (parallel system), the system's reliability is always larger than the components' reliability if $n > 1$. If $k = n$ (series system), the system's reliability is always less than the components' reliability. However, when $1 < k < n$, the system's reliability is less than p when p is small and greater than p when p is large. There exists a break-even point when $R(n; k) = p$. This is illustrated in Figure 4.2. However, if p is close to 1, improving p would be very difficult and would have little improvement on the system's reliability.

The decrease of k will lead to a larger system reliability. But the value of k generally cannot be determined by the reliability analyst. The value k is normally determined by the characteristics of the system.

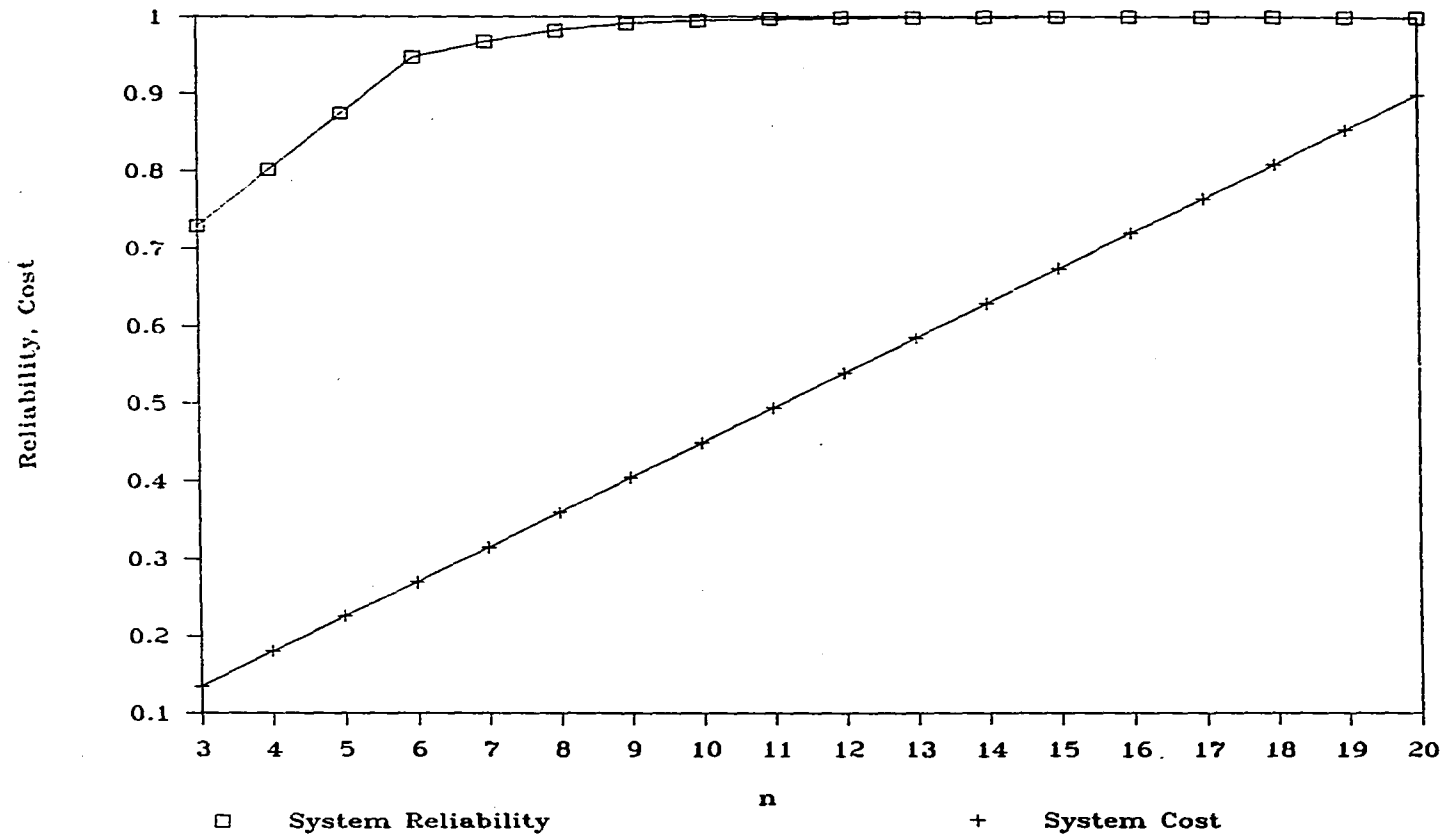


Figure 4.1: k -out-of- n :G system reliability and cost as functions of n with $p = 0.9$, $k = 3$, and $\text{Cost} = 0.045n$

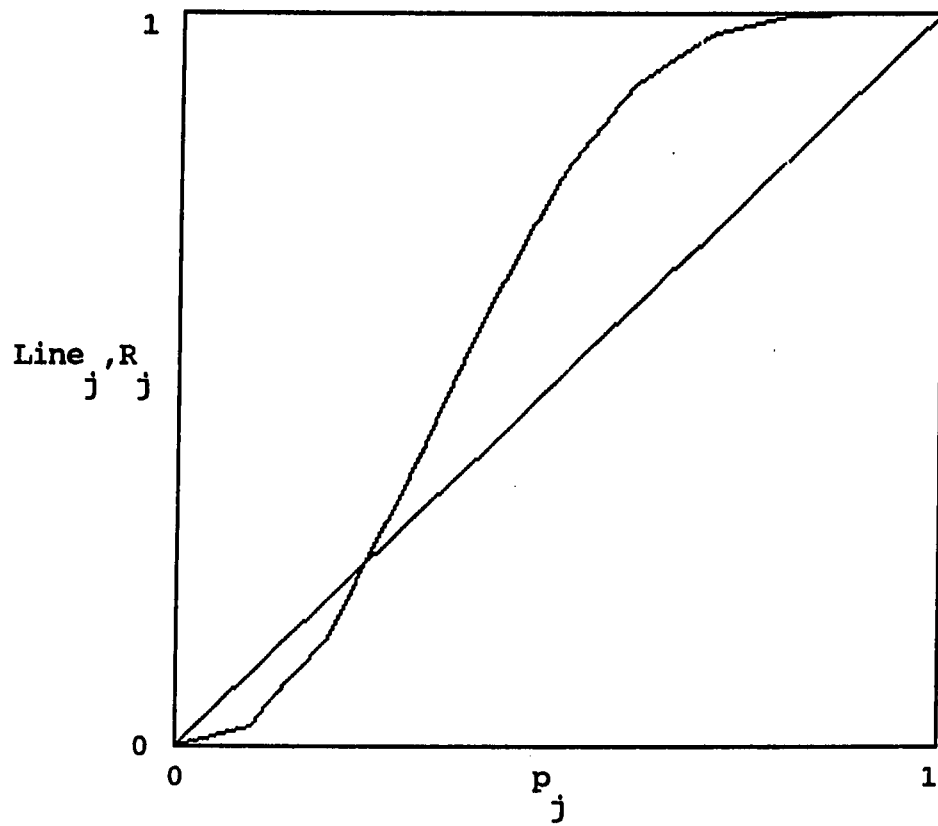


Figure 4.2: k -out-of- n :G system reliability as a function of p with $n = 7$, $k = 3$, and $\text{Line} = p$

Consecutive- k -out-of- n Systems

Special reliability formulas

The following reliability formulas for special consecutive- k -out-of- n systems are provided. They are simple and useful for hand calculations and especially for the proofs of some optimal design results to be presented later in this dissertation.

Lemma 3 *The reliability of a linear consecutive- k -out-of- n : G system with $n \leq 2k$ is:*

$$R_{G'}(n; k) = \sum_{i=1}^{n-k+1} \left(q_{i+k} \prod_{j=i}^{i+k-1} p_j \right) \quad (4.10)$$

where $q_{n+1} \equiv 1$, or

$$R_G(n; k) = \sum_{i=1}^{n-k+1} \left(\prod_{j=i}^{i+k-1} p_j \right) - \sum_{i=1}^{n-k} \left(\prod_{j=i}^{i+k} p_j \right) \quad (4.11)$$

Proof of Lemma 3

Using Lemma 2 in Kuo, Zhang, and Zuo [49] and keeping in mind that n is not greater than $2k$, we have

$$\begin{aligned} R_{G'}(n; k) &= R(n-1; k) - Q(n-k-1; k)q_{n-k} \left(\prod_{i=n-k+1}^n p_i \right) \\ &= R((2, n); k) + Q((k-2, n); k)q_{k+1} \left(\prod_{i=1}^k p_i \right) \\ &= R((2, n); k) + q_{k+1} \left(\prod_{i=1}^k p_i \right) \\ &= R((3, n); k) - q_{k+2} \left(\prod_{i=2}^{k+1} p_i \right) + q_{k+1} \left(\prod_{i=1}^k p_i \right) \\ &= \dots \end{aligned}$$

$$\begin{aligned}
&= R((n-k+1, n); k) + \sum_{i=1}^{n-k} \left(q_{i+k} \prod_{j=i}^{i+k-1} p_j \right) \\
&= p_{n-k+1} \cdots p_n + \sum_{i=1}^{n-k} \left(q_{i+k} \prod_{j=i}^{i+k-1} p_j \right) \\
&= \sum_{i=1}^{n-k+1} \left(q_{i+k} \prod_{j=i}^{i+k-1} p_j \right), \quad \text{if } q_{n+1} = 1.
\end{aligned}$$

(Q.E.D.)

Lemma 4 *The unreliability of a linear consecutive- k -out-of- n : F system with $n \leq 2k$ is:*

$$Q_{F(n; k)} = \sum_{i=1}^{n-k+1} \left(p_{i+k} \prod_{j=i}^{i+k-1} q_j \right) \quad (4.12)$$

where $p_{n+1} \equiv 1$, or

$$Q_{F(n; k)} = \sum_{i=1}^{n-k+1} \left(\prod_{j=i}^{i+k-1} q_j \right) - \sum_{i=1}^{n-k} \left(\prod_{j=i}^{i+k} q_j \right). \quad (4.13)$$

Proof of Lemma 4

A proof is immediate with Lemma 3 above and Lemma 1 of Kuo, Zhang, and Zuo [49]. (Q.E.D.)

Lemma 5 *The reliability of a circular consecutive- k -out-of- n : G system with $n \leq 2k+1$ is:*

$$R_{C'G(n; k)} = \sum_{i=1}^n \left(q_{i+k} \prod_{j=i}^{i+k-1} p_j \right) + \prod_{i=1}^n p_i \quad (4.14)$$

or,

$$R_{C'G(n; k)} = \sum_{i=1}^n \left(\prod_{j=i}^{i+k-1} p_j \right) - \sum_{i=1}^n \left(\prod_{j=i}^{i+k} p_j \right) + \prod_{i=1}^n p_i \quad (4.15)$$

where $p_j = p_{j-n}$ if $j > n$.

Proof of Lemma 5

For $n \leq k$, the problem is trivial. If $n = k + 1$, using Lemma 3 in Kuo, Zhang, and Zuo [49], we have

$$\begin{aligned}
 R_{C'G}(k+1; k) &= \sum_{i=1}^{k+1} \left(\prod_{j=i+1}^{i+k} p_j \right) - k \left(\prod_{i=1}^{k+1} p_i \right) \\
 &= \sum_{i=1}^{k+1} \left(\prod_{j=i+1}^{i+k} p_j - \prod_{j=1}^{k+1} p_j \right) + \prod_{i=1}^{k+1} p_i \\
 &= \sum_{i=1}^{k+1} q_{i+k} \left(\prod_{j=i}^{i+k-1} p_j \right) + \prod_{i=1}^{k+1} p_i.
 \end{aligned}$$

Use mathematical induction to prove Lemma 5. Assume Formula (4.14) is correct for l ($k < l < 2k + 1$), i.e.,

$$R_{C'G}(l; k) = \sum_{i=1}^l \left(q_{i+k} \prod_{j=i}^{i+k-1} p_j \right) + \prod_{i=1}^l p_i, \quad (4.16)$$

where $p_j = p_{j-l}$ if $j > l$.

Let $n = l + 1$, then using Lemma 3 of Kuo, Zhang, and Zuo [49], Formula (4.16) and Lemma 3 above, we have

$$\begin{aligned}
 R_{C'G}(l+1; k) &= \sum_{i=1}^k \left(q_i q_{l-k+i} \left(\prod_{j=1}^{i-1} p_j \right) \left(\prod_{j=l-k+i+1}^{l+1} p_j \right) \right) \\
 &\quad + q_{l+1} R_G(l; k) + p_{l+1} R_{C'G}(l; k) \\
 &= \sum_{i=1}^k \left(q_i q_{l-k+i} \left(\prod_{j=1}^{i-1} p_j \right) \left(\prod_{j=l-k+i+1}^{l+1} p_j \right) \right) \\
 &\quad - q_{l+1} \left(\sum_{i=1}^{l-k+1} \left(q_{i+k} \prod_{j=i}^{i+k-1} p_j \right) \right) \\
 &\quad + p_{l+1} \left(\sum_{i=1}^l q_{i+k} \left(\prod_{j=i}^{i+k-1} p_j \right) + \prod_{i=1}^l p_i \right)
 \end{aligned}$$

$$\begin{aligned}
&= (q_1 q_{l-k-1} p_{l-k-2} \cdots p_{l+1} \\
&\quad + q_2 q_{l-k+2} p_{l-k+3} \cdots p_{l+1} p_1 \\
&\quad + \cdots + q_k q_l p_{l+1} p_1 \cdots p_{k-1}) \\
&\quad + (q_{l+1} q_{k+1} p_1 \cdots p_k + q_{l+1} q_{k+2} p_2 \cdots p_{k+1} \\
&\quad + \cdots + q_{l+1} q_l p_{l-k} \cdots p_{l-1} + q_{l+1} p_{l-k+1} \cdots p_l) \\
&\quad + (p_{l+1} q_{k+1} p_1 \cdots p_k + p_{l+1} q_{k+2} p_2 \cdots p_{k+1} \\
&\quad + \cdots + p_{l+1} q_l p_{l-k} \cdots p_{l-1} + p_{l+1} q_1 p_{l-k+1} \cdots p_l \\
&\quad + \cdots + p_{l+1} q_k p_l p_1 \cdots p_{k-1}) + p_1 \cdots p_{l+1} \\
&= \sum_{i=1}^n \left(q_{i+k} \prod_{j=i}^{i+k-1} p_j \right) + \prod_{i=1}^n p_i.
\end{aligned}$$

(Q.E.D.)

Lemma 6 *The unreliability of a circular consecutive- k -out-of- $n:F$ system with $n \leq 2k+1$ is:*

$$Q_{CF}(n; k) = \sum_{i=1}^n \left(p_{i+k} \prod_{j=i}^{i+k-1} q_j \right) + \prod_{i=1}^n q_i \quad (4.17)$$

or,

$$Q_{CF}(n; k) = \sum_{i=1}^n \left(\prod_{j=i}^{i+k-1} q_j \right) - \sum_{i=1}^n \left(\prod_{j=i}^{i+k} q_j \right) + \prod_{i=1}^n q_i \quad (4.18)$$

where $q_j = q_{j-n}$ if $j > n$.

Proof of Lemma 6

The result is immediate with Lemma 5 above and Lemma 1 of Kuo, Zhang, and Zuo [49]. (Q.E.D.)

Disjoint minimal path method

As we have seen previously, minimal paths and cuts may be used to calculate the reliability of a k -out-of- n system. The complexity of the SDP method is a function of the number of minimal paths or minimal cuts, and the complexity of finding all the disjoint minimal paths. The SDP method may be efficient for consecutive- k -out-of- n systems because the components are ordered in such systems. Unlike the k -out-of- n systems, where k -out-of- $n:F$ and $(n - k + 1)$ -out-of- $n:G$ systems are equivalent, a consecutive- k -out-of- $n:F$ system is not equivalent to a consecutive- $(n - k + 1)$ -out-of- $n:G$ system. In reliability evaluation, it is easier to work with minimal paths for consecutive- k -out-of- $n:G$ systems and minimal cuts for consecutive- k -out-of- $n:F$ systems.

Let us consider a consecutive- k -out-of- $n:G$ system and define

$U \equiv$ the set of certainty

$P_n \equiv$ the set of disjoint minimal path sets when the system size is n

$P_n = \Phi$, if $n < k$.

Then,

$$\begin{aligned}
 P_k &= \{(1, 2, \dots, k)\} \\
 P_{k+1} &= \{(1, 2, \dots, k), (\bar{1}, 2, 3, \dots, k, k+1)\} = \{P_k, (\bar{1}, 2, 3, \dots, k, k+1)\} \\
 P_{k+2} &= \{(1, 2, \dots, k), (\bar{1}, 2, 3, \dots, k, k+1), (\bar{2}, 3, \dots, k+1, k+2)\} \\
 &= \{P_{k+1}, (\bar{2}, 3, \dots, k+1, k+2)\} \\
 P_{2k} &= \{(1, 2, \dots, k), (\bar{1}, 2, 3, \dots, k+1), (\bar{2}, 3, 4, \dots, k+2), \\
 &\quad \dots, (\bar{k}, k+1, \dots, 2k)\}
 \end{aligned}$$

$$\begin{aligned}
&= \{P_{2k-1}, (\bar{k}, k+1, \dots, 2k)\} \\
P_{2k+1} &= \{(1, 2, \dots, k), (\bar{1}, 2, 3, \dots, k+1), (\bar{2}, 3, 4, \dots, k+2), \\
&\quad \dots, (\bar{k}, k+1, \dots, 2k), (U - P_k)(\overline{k+1}, k+2, \dots, 2k+1)\} \\
&= \{P_{2k}, (U - P_k)(\overline{k+1}, k+2, \dots, 2k+1)\}.
\end{aligned}$$

In general,

$$P_n = \{P_{n-1}, (U - P_{n-k-1})(\overline{n-k}, n-k+1, \dots, n)\}. \quad (4.19)$$

Define

$$\Delta_n \equiv P_n - P_{n-1}, \quad (4.20)$$

then,

$$\Delta_n = (U - P_{n-k-1})(\overline{n-k}, n-k+1, \dots, n). \quad (4.21)$$

In fact, Δ_n is the new disjoint minimal path added to the original system by introducing a component, n , into the $(n-1)$ -component system. $\Pr\{\Delta_n\}$ is the incremental reliability to the system by increasing the system size from $n-1$ to n .

With the disjoint minimal paths available, adding the probabilities of the members of P_n will directly give the reliability of a consecutive- k -out-of- n :G system:

$$R(n; k) = R(n-1; k) + (1 - R(n-k-1; k))q_{n-k}p_{n-k+1} \cdots p_n. \quad (4.22)$$

This is the same formula found by Kuo, Zhang, and Zuo [49]. Thus, we have presented another way of developing the reliability formula for a linear consecutive- k -out-of- n :G system.

Component importances

Using the component reliability importance definition given by Birnbaum [9], Papastavridis [66], and Kuo, Zhang, and Zuo [49] provided the component reliability importance functions of a consecutive- k -out-of- n :F system and a consecutive- k -out-of- n :G system, respectively.

$$I_i = \frac{R(i-1; k)R'(n-i; k) - R(n; k)}{q_i}, \quad \text{for F systems,} \quad (4.23)$$

$$I_i = \frac{1}{p_i} [R(n; k) - R(i-1; k) - R'(n-i; k) + R(i-1; k)R'(n-i; k)], \quad \text{for G systems.} \quad (4.24)$$

The following lemmas describe the component reliability importance patterns of a linear consecutive-2-out-of- n :F system and a linear consecutive-2-out-of- n :G system.

Lemma 7 *In a linear consecutive-2-out-of- n :F system with i.i.d. components and $n > 4$, the component reliability importance function, I_i , of position i is symmetric to $i = (n+1)/2$, i.e., $I_i = I_{n-i+1}$, and satisfies the following conditions.*

$$\begin{aligned} I_{2i} &> I_{2i-1}, & \text{for } 2i \leq (n+1)/2 \\ I_{2i} &< I_{2(i-1)}, & \text{for } 2i \leq (n+1)/2 \\ I_{2i+1} &> I_{2i-1}, & \text{for } 2i+1 \leq (n+1)/2 \end{aligned}$$

Proof of Lemma 7

In the following derivations, we will use notation $R(i)$ to represent $R(i; 2)$. From Hwang [40], we have the following for a linear consecutive-2-out-of- n :F system with i.i.d. components:

$$R(n+1) = R(n) - R(n-2)pq^2$$

$$R(n-1) = R(n) - R(n-3)pq^2$$

$$R(0) = 1$$

$$R(1) = 1$$

$$R(i) > R(i+1), \text{ for } i \geq 1.$$

From Papastavridis [66], we have

$$I_i = \frac{R(i-1)R'(n-i) - R(n)}{q_i}.$$

Define

$$J_i = q_i I_i + R(n) = R(i-1)R'(n-i).$$

When all the components are i.i.d., we have

$$R(i) = R'(i), \text{ for all } i,$$

$$J_i = R(i-1)R(n-i).$$

Since q_i and $R(n)$ are constant to position i in a consecutive-2-out-of- n :F system with i.i.d. components, I_i and J_i have the same pattern. We will only consider J_i from now on.

$$R(m) > pR(m-1), \text{ for } m > 0, 0 < p < 1 \quad (4.25)$$

because

$$\begin{aligned} R(m) &= p_m \Pr[\text{the } m \text{ component system works} | \text{the } m\text{th works}] \\ &\quad + q_m \Pr[\text{the } m \text{ component system works} | \text{the } m\text{th fails}] \end{aligned}$$

$$\begin{aligned}
&= p_m R(m-1) \\
&\quad - q_m \Pr[\text{the } m \text{ component system work} \mid \text{the } m\text{th fails}] \\
&> p_m R(m-1), \text{ for } m > 1, 0 < p < 1.
\end{aligned}$$

Another general result is

$$R(m) < R(i)R(m-i), \quad \text{for } i = 1, 2, \dots, m-1 \quad (4.26)$$

because

$$R(m) = \begin{cases} R(1)R(m-1) - pq^2 R(m-3), \\ \quad \text{for } i = 1 \text{ or } m-1 \\ R(i)R(m-i) - p^2 q^2 R(i-2)R(m-i+2), \\ \quad \text{for } i = 2, \dots, m-2 \end{cases}.$$

(1) J_1 :

$$J_1 = R(0)R(n-1) = R(n-1).$$

(2) J_2 :

$$J_2 = R(1)R(n-2) = R(n-2) > R(n-1) = J_1, \text{ for } n \geq 3.$$

(3) J_3 :

$$R(n-1) = \begin{cases} R(1)R(n-2) - pq^2 R(n-4), \\ R(i)R(n-1-i) - p^2 q^2 R(n-3-i)R(i-2), \text{ for } i > 1 \end{cases}. \quad (4.27)$$

As a result,

$$J_2 = R(1)R(n-2) = R(n-1) + pq^2 R(n-4) \quad (4.28)$$

$$\begin{aligned}
J_3 &= R(2)R(n-3) = R(n-1) + p^2 q^2 R(n-5) \\
&= J_1 + p^2 q^2 R(n-5).
\end{aligned} \quad (4.29)$$

By letting $m = n - 4$ in Equation (4.25), we have

$$R(n - 4) > pR(n - 5), \text{ for } n \geq 5.$$

Therefore, from Equations (4.28) and (4.29):

$$J_3 < J_2, \text{ for } n \geq 5$$

$$J_3 > J_1, \text{ for } n \geq 5.$$

(4) J_4 : Using Formula (4.27),

$$J_2 = R(1)R(n - 2) = R(n - 1) + pq^2R(n - 4)$$

$$J_4 = R(3)R(n - 4) = R(n - 1) + p^2q^2R(1)R(n - 6).$$

It is true that

$$J_4 < J_2.$$

because

$$\begin{aligned} & R(n - 4) \\ &= p_{n-5}R(1)R(n - 6) \\ &+ q_{n-5} \Pr[\text{the } n - 4 \text{ component system works} \mid \text{the } (n - 5)\text{th fails}] \\ &> pR(1)R(n - 6). \end{aligned}$$

(5) Comparing J_{m+1} with J_m :

Assuming $m + 3 \leq (n + 1)/2$ and using Formula (4.27):

$$J_{m+3} = R(m + 2)R(n - m - 3)$$

$$\begin{aligned}
&= R(n-1) + p^2 q^2 R(m)R(n-m-5) \\
J_{m+2} &= R(m+1)R(n-m-2) \\
&= R(n-1) + p^2 q^2 R(m-1)R(n-m-4) \\
J_{m+1} &= R(m)R(n-m-1) \\
J_m &= R(m-1)R(n-m).
\end{aligned}$$

If $J_{m+1} > J_m$, then we have the following:

$$\begin{aligned}
R(m)R(n-m-1) &> R(m-1)R(n-m), \quad (n-m-4) \\
R(m)R(n-4-m-1) &> R(m-1)R(n-4-m) \\
R(m)R(n-m-5) &> R(m-1)R(n-m-4) \\
R(m+2)R(n-m-3) &> R(m+1)R(n-m-2) \\
J_{m+3} &> J_{m+2}.
\end{aligned}$$

If $J_{m+1} < J_m$, then we have the following:

$$\begin{aligned}
R(m)R(n-m-1) &< R(m-1)R(n-m), \quad (n-m-4) \\
R(m)R(n-4-m-1) &< R(m-1)R(n-4-m) \\
R(m)R(n-m-5) &< R(m-1)R(n-m-4) \\
R(m+2)R(n-m-3) &< R(m+1)R(n-m-2) \\
J_{m+3} &< J_{m+2}.
\end{aligned}$$

(6) Comparing J_{m+2} with J_m :

Assuming $m+4 \leq (n+1)/2$ and using Formula (4.27):

$$\begin{aligned}
J_{m+4} &= R(m-3)R(n-m-4) \\
&= R(n-1) + p^2 q^2 R(m+1)R(n-m-6) \\
J_{m+2} &= R(m+1)R(n-m-2) \\
&= R(n-1) + p^2 q^2 R(m-1)R(n-m-4) \\
J_m &= R(m-1)R(n-m) \\
&= R(n-1) + p^2 q^2 R(m-3)R(n-m-2).
\end{aligned}$$

If $J_{m+2} > J_m$, then we have the following:

$$\begin{aligned}
R(m+1)R(n-m-2) &> R(m-1)R(n-m), \quad (n \leftarrow n-4) \\
R(m+1)R(n-4-m-2) &> R(m-1)R(n-4-m) \\
R(m+1)R(n-m-6) &> R(m-1)R(n-m-4) \\
R(m+3)R(n-m-4) &> R(m+1)R(n-m-2) \\
J_{m+4} &> J_{m+2}.
\end{aligned}$$

If $J_{m+1} < J_m$, then we have the following:

$$\begin{aligned}
R(m+1)R(n-m-2) &< R(m-1)R(n-m), \quad (n \leftarrow n-4) \\
R(m+1)R(n-4-m-2) &< R(m-1)R(n-4-m) \\
R(m+1)R(n-m-6) &< R(m-1)R(n-m-4) \\
R(m+3)R(n-m-4) &< R(m+1)R(n-m-2) \\
J_{m+4} &< J_{m+2}.
\end{aligned}$$

Lemma 8 *The component reliability importance function, I_i , in a consecutive-2-out-of- n :G system has the same pattern as the component reliability importance function in a consecutive-2-out-of- n :F system, given that the components are i.i.d.*

Proof of Lemma 8

From the proof of Lemma 7, we know the following for a consecutive-2-out-of- n :F system:

$$\begin{aligned}
 R(n+1) &= R(n) - R(n-2)pq^2 \\
 R(n-1) &= R(n) + R(n-3)pq^2 \\
 R(0) &= 1 \\
 R(1) &= 1 \\
 R(i) &> R(i+1), \text{ for } i \geq 1 \\
 I_i &= \frac{R(i-1)R'(n-i) - R(n)}{q_i}.
 \end{aligned}$$

Using the results in Kuo, Zhang, and Zuo [49], we have the following for a consecutive-2-out-of- n :G system:

$$\begin{aligned}
 Q(n+1) &= Q(n) - Q(n-2)qp^2 \\
 Q(n-1) &= Q(n) + Q(n-3)qp^2 \\
 Q(0) &= 1 \\
 Q(1) &= 1 \\
 Q(i) &> Q(i+1), \text{ for } i \geq 1 \\
 I_i &= \frac{Q(i-1)Q'(n-i) - Q(n)}{p_i}.
 \end{aligned}$$

As a result, all arguments for $R(i)$, q_i , and I_i of the consecutive-2-out-of- n :F system in the Proof of Lemma 8 hold for $Q(i)$, p_i , and I_i of the consecutive-2-out-of- n :G system here. Thus, the same component reliability pattern holds for the consecutive-2-out-of- n :G system.

Figure 4.3 shows the position importance pattern of a consecutive-2-out-of-20:G system with $p = 0.5$. From the position importance patterns, it is clear that even though the component reliabilities are the same, some positions are more important than others. A very intuitive way of optimal design is to assign more reliable components to more important positions. Such a heuristic is implemented in Chapter 5.

Strict Consecutive- k -out-of- n Systems

The author does not think that the concept of a strict consecutive- k -out-of- n :F system given by Bollinger [13] is valid (see review in Chapter 3). In fact, in a normal consecutive- k -out-of- n :F system, it is assumed that the system is working even if there are many failure strings of length longer than zero, as long as they are shorter than k . These failure strings may degrade the performance of the system, but the system can still perform the task for which it is designed. If the maintenance of the system is efficient, these failure strings shorter than k can be easily eliminated before they accumulate and result in a system failure. These arguments agree with Bollinger [13]. However, the strings shorter than k but longer than zero have nothing to do with the failure probability of the system. The reason is that the system fails whenever there is at least one failure string of length greater than $k - 1$ no matter whether there are

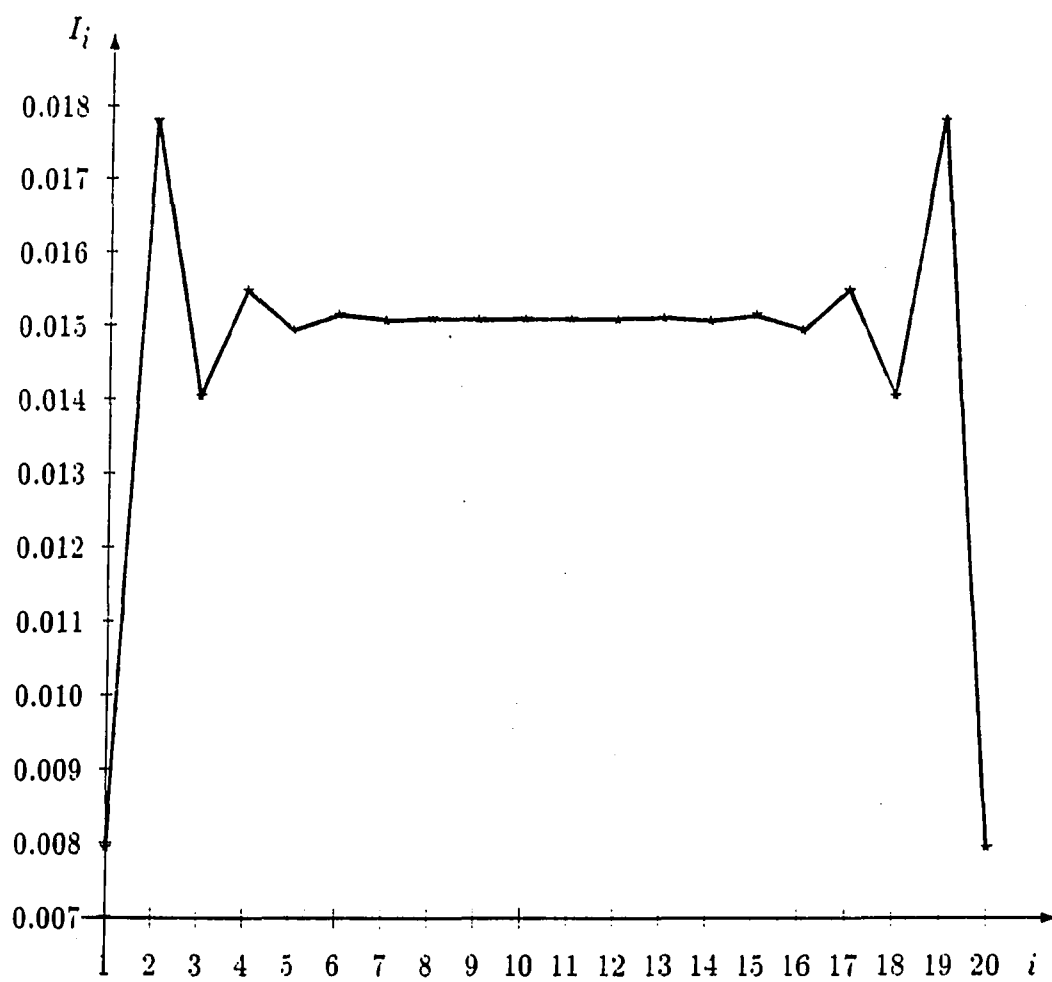


Figure 4.3: The pattern of component reliability importance of a consecutive-2-out-of-20:G system

other failure strings shorter than k or not. As a result, the failure probability of the system is the probability that there is at least one failure string of length at least k . The condition that there are no isolated failure strings of fewer than k consecutive components should not be added to it. The definition of failure probability given in Bollinger [13] underestimates the failure probability of the system.

If the definition given in [13] is used, the reliability of the consecutive- k -out-of- n :F system is not equal to 1 minus the failure probability of the system. The case FOFOFFOFFF ($k = 3$) is not included in the failure probability calculation. It cannot be included in the system's reliability calculation either because certainly it does not result in an operating system. Thus, the concept of strict consecutive- k -out-of- n :F system is useless.

Summary

This chapter studied some characteristics of the k -out-of- n system. Special reliability evaluation formulas for consecutive- k -out-of- n systems were developed. Comments were made on the concept of the strict consecutive- k -out-of- n :F systems. The component reliability importance pattern of a linear consecutive-2-out-of- n system was identified.

CHAPTER 5. OPTIMAL SYSTEM DESIGN OF CONSECUTIVE- k -OUT-OF- n SYSTEMS

Introduction

Many researchers have studied reliability evaluation of the consecutive- k -out-of- n systems. However, not much has been reported on the optimal design issue of such systems, especially of circular systems.

The system design of a consecutive- k -out-of- n system is to arrange components with different reliabilities to different positions such that system reliability is maximized. In some cases only the ranking of the components' reliabilities determines the optimal arrangements of the components, while in other cases exact component reliabilities must be available in order to determine the optimal arrangement of the components. If a system design depends solely on the ranking of the component reliabilities, it is called invariant system design. Thus, efforts have been made by many researchers to find invariant optimal designs of some consecutive- k -out-of- n systems.

This chapter is devoted to the optimal design of both linear and circular consecutive- k -out-of- n systems. Theorems and lemmas are provided either to find the invariant optimal designs of the systems, or to prove that invariant optimal designs do not exist. All cases of consecutive- k -out-of- n systems are theoretically analyzed.

A heuristic method and a randomization method are provided to find at least suboptimal designs of a consecutive- k -out-of- n systems. A special binary search method is presented also.

We assume that p_1, p_2, \dots, p_n are all positive and distinct, since other cases can be viewed as limits of this case. Without this assumption, some strict inequalities proven below will become non-strict and consequently, the optimal design to be presented is unique up to equivalent components (components with the same reliability). We also treat the reversed sequence of a system configuration be the same as the original.

Invariant Optimal System Designs

Theorem 1 *A necessary condition for the optimal design of a circular consecutive- k -out-of- $n:F$ system with $n = k + 2$ is*

$$(q_i - q_j)(q_{i-1} - q_{j+1}) < 0, \quad \text{for } j = i + 1, i + 2 \quad (5.1)$$

$$(q_i - q_j) \left[q_{i+1} q_{j-1} (q_{i-1} - q_{j+1}) + q_{i-1} q_{j+1} (q_{i+1} - q_{j-1}) \right] < 0, \quad \text{for } j \geq i + 3 \quad (5.2)$$

where i ranges from 1 through n , q_i is the unreliability of the component at position i , and $q_j = q_{j-n}$ if $j > n$.

Proof of Theorem 1

The formula for the unreliability of a circular consecutive- k -out-of- $n:F$ system given in Lemma 6 is used in the following proof.

- (1) For the case $j = i + 1$, a proof is given in Kuo and Zhang [48].

(2) When $j = i - 2$, interchanging the component at position i (q_i) and the component at position j (q_j) results in the following change in system unreliability

$$\begin{aligned}
 \Delta &= Q_{C'F(n;k)}^{\text{before}} - Q_{C'F(n;k)}^{\text{after}} \\
 &= [q_j q_{j+1} \cdots q_{i-1} - q_{i+1} q_j q_{j+1} \cdots q_{i-2} + q_{j+2} \cdots q_{i-1} q_i q_{i+1} - \\
 &\quad q_{j+1} \cdots q_{i-1} q_i - q_{i+1} q_j \cdots q_{i-1} - q_{j-1} \cdots q_i q_{i+1}] \\
 &\quad - [q_i q_{j+1} \cdots q_{i-1} + q_{i+1} q_i q_{j+1} \cdots q_{i-2} + q_{j+2} \cdots q_{i-1} q_j q_{i+1} + \\
 &\quad q_{j+1} \cdots q_{i-1} q_j - q_{i+1} q_i \cdots q_{i-1} - q_{j-1} \cdots q_j q_{i+1}] \\
 &= (q_j - q_i) q_{i+1} q_{j+1} \cdots q_{i-2} - (q_j - q_i) q_{i+1} q_{j+2} \cdots q_{i-1} \\
 &= (q_j - q_i) (q_{j+1} - q_{i-1}) q_{i+1} q_{j+2} \cdots q_{i-2} \\
 &= (q_i - q_j) (q_{i-1} - q_{j+1}) q_{i+1} q_{j+2} \cdots q_{i-2}.
 \end{aligned}$$

If $q_i < q_j$ and $q_{i-1} > q_{j+1}$, then $\Delta < 0$. In other words, if $(q_i - q_j)(q_{i-1} - q_{j+1}) < 0$, system unreliability can only be increased (system reliability can only be decreased) by interchanging components i and $i+2$. As a result, if the system is already optimally designed, the following inequality must be true

$$(q_i - q_j)(q_{i-1} - q_{j+1}) < 0.$$

(3) When $j \geq i + 3$, interchanging the component at position i (q_i) and the component at position j (q_j) results in the following change in system unreliability

$$\begin{aligned}
 \Delta &= Q_{C'F(n;k)}^{\text{before}} - Q_{C'F(n;k)}^{\text{after}} \\
 &= [q_{j+2} \cdots q_{i-1} q_i q_{i+1} \cdots q_{j-1} - q_{j+1} \cdots q_{i-1} q_i q_{i+1} \cdots q_{j-2} \\
 &\quad + q_{i+2} \cdots q_{j-1} q_j q_{j+1} \cdots q_{i-1} + q_{i+1} \cdots q_{j-1} q_j q_{j+1} \cdots q_{i-2}
 \end{aligned}$$

$$\begin{aligned}
& -q_{j+1} \cdots q_{i-1} q_i q_{i+1} \cdots q_{j-1} - q_{i+1} \cdots q_{j-1} q_j q_{j+1} \cdots q_{i-1} \\
& -[q_{j+2} \cdots q_{i-1} q_j q_{i+1} \cdots q_{j-1} + q_{j+1} \cdots q_{i-1} q_j q_{i+1} \cdots q_{j-2} \\
& + q_{i+2} \cdots q_{j-1} q_i q_{j+1} \cdots q_{i-1} + q_{i+1} \cdots q_{j-1} q_i q_{j+1} \cdots q_{i-2} \\
& - q_{j+1} \cdots q_{i-1} q_j q_{i+1} \cdots q_{j-1} - q_{i+1} \cdots q_{j-1} q_i q_{j+1} \cdots q_{i-1}] \\
& = (q_i - q_j)[q_{i+1} q_{j-1} (q_{i-1} - q_{j+1}) + q_{i-1} q_{j+1} (q_{i+1} - q_{j-1})] \times \\
& \quad q_{i+2} \cdots q_{j-2} q_{j+2} \cdots q_{i-2}.
\end{aligned}$$

The interchange of components at positions i and j can only increase the system's unreliability if $(q_i - q_j)[q_{i+1} q_{j-1} (q_{i-1} - q_{j+1}) + q_{i-1} q_{j+1} (q_{i+1} - q_{j-1})] < 0$.

In summary, if a system is optimally designed, then the conditions listed in Theorem 1 must be true. As a result, the conditions in Theorem 1 are necessary conditions for the optimal design of a circular consecutive- k -out-of- n :F system with $n = k + 2$. (Q.E.D.)

Lemma 9 *A stronger necessary condition than that in Theorem 1 for the optimal design of a circular consecutive- k -out-of- n :F system with $n = k + 2$ is*

$$(q_i - q_j)(q_{i-1} - q_{j+1}) < 0, \quad \text{for } j > i. \quad (5.3)$$

where i ranges from 1 through n , q_i is the unreliability of the component at position i , and $q_j = q_{j-n}$ if $j > n$.

Proof of Lemma 9

When $j = i + 1$ or $j = i + 2$, the condition in Lemma 9 is equivalent to that in Theorem 1. Now look at the condition for $j > i + 2$ in Theorem 1,

$$(q_i - q_j)[q_{i+1} q_{j-1} (q_{i-1} - q_{j+1}) + q_{i-1} q_{j+1} (q_{i+1} - q_{j-1})] < 0$$

or equivalently,

$$(q_i - q_j)(q_{i-1} - q_{j+1})q_{i+1}q_{j-1} + (q_i - q_j)(q_{i-1} - q_{j-1})q_{i-1}q_{j+1} < 0. \quad (5.4)$$

The left hand side of the above inequality is a sum of two terms. We can prove the second term is always negative if the first term is negative for all i .

Assume $(q_i - q_j)(q_{i-1} - q_{j+1}) < 0$ for $j > i$, then for j incremented by 1 ($j' = j + 1$) and i decremented by 1 ($i' = i - 1$), the second term on the left hand side of inequality (5.4) becomes

$$(q_{i'} - q_{j'})(q_{i'+1} - q_{j'-1})q_{i'-1}q_{j'+1},$$

or

$$(q_{i-1} - q_{j+1})(q_i - q_j)q_{i-2}q_{j+2}$$

and this is negative from the assumption.

As a result, if the condition in Lemma 9 is satisfied, the conditions in Theorem 1 must be satisfied. The condition in Lemma 9 is a stronger necessary condition for the optimal design of a circular consecutive- k -out-of- $n:F$ with $n = k + 2$. (Q.E.D.)

Theorem 2 *For a circular consecutive- k -out-of- $n:F$ system with $n = k + 2$, the only configuration satisfying the necessary condition in Lemma 9 is*

$$C_n = (1, n-1, 3, n-3, 5, n-5, \dots, n-6, 6, n-4, 4, n-2, 2, n, 1). \quad (5.5)$$

Thus, C_n is the invariant optimal design of a circular consecutive- k -out-of- $n:F$ system with $n = k + 2$.

Proof of Theorem 2

To satisfy the necessary condition specified in Lemma 9, component 1 must be adjacent to components n and $n-1$. If not, say, 1 is not adjacent to $n-1$, but to some $i, i < n-1$. Let j be the item following $n-1$ in the sequence $1, i, \dots, n-1, j$. This sequence violates the condition specified in Lemma 9, since $q_1 > q_j$ and $q_i > q_{n-1}$. Similarly, we can show that n must be adjacent to 1 and 2, 2 must be adjacent to n and $n-2$, and so on. In essence, C'_n is the only configuration satisfying the necessary condition specified in Lemma 9. Thus, it is the invariant optimal design of a circular consecutive- k -out-of- n :F system with $n = k + 2$. (Q.E.D.)

Theorem 3 *A necessary condition for the optimal design of a circular consecutive- k -out-of- n :G system with $k < n \leq 2k + 1$ is*

$$(p_i - p_j)(p_{i-1} - p_{j+1}) > 0, \quad j > i. \quad (5.6)$$

where i ranges from 1 through n , p_i is the reliability of the component at position i , and $p_j = p_{j-n}$ if $j > n$.

Proof of Theorem 3

The formula of the reliability of a circular consecutive- k -out-of- n :G system given in Lemma 5 is used in the following proof.

Let $i < k$ (proof for $i \geq k$ is similar) and $j > i$. Interchanging the component at position i (p_i) and the component at position j (p_j) results in the following change in system reliability

$$\Delta = R_{C'G(n;k)}^{\text{before}} - R_{C'G(n;k)}^{\text{after}}$$

$$\begin{aligned}
&= p_{n-k+i+1} \cdots p_n p_1 \cdots p_{i-1} (p_i - p_j) \\
&\quad + p_{n-k+i+2} \cdots p_n p_1 \cdots p_{i-1} (p_i - p_j) p_{i+1} \\
&\quad + \cdots + p_{n-k+j} \cdots p_n p_1 \cdots p_{i-1} (p_i - p_j) p_{i+1} \cdots p_{j-1} \\
&\quad - (p_j - p_i) p_{j+1} \cdots p_{j+k-1} + p_{j-1} (p_j - p_i) p_{j+1} \cdots p_{j+k-2} \\
&\quad + \cdots + p_{i+1} \cdots p_{j-1} (p_j - p_i) p_{j+1} \cdots p_{k+i} \\
&\quad - p_{n-k+i} \cdots p_n p_1 \cdots p_{i-1} (p_i - p_j) \\
&\quad - p_{n-k+i+1} \cdots p_n p_1 \cdots p_{i-1} (p_i - p_j) p_{i+1} \\
&\quad - \cdots - p_{n-k+j-1} \cdots p_n p_1 \cdots p_{i-1} (p_i - p_j) p_{i+1} \cdots p_{j-1} \\
&\quad - (p_j - p_i) p_{j+1} \cdots p_{j+k} - p_{j-1} (p_j - p_i) p_{j+1} \cdots p_{j+k-1} \\
&\quad - \cdots - p_{i+1} \cdots p_{j-1} (p_j - p_i) p_{j+1} \cdots p_{k+i+1} \\
&= (p_i - p_j) (q_{n-k+i} p_{n-k+i+1} \cdots p_n p_1 \cdots p_{i-1} \\
&\quad + q_{n-k+i+1} p_{n-k+i+2} \cdots p_n p_1 \cdots p_{i-1} p_{i+1} \\
&\quad + \cdots + q_{n-k+j-1} p_{n-k+j} \cdots p_n p_1 \cdots p_{i-1} p_{i+1} \cdots p_{j-1}) \\
&\quad + (p_j - p_i) (p_{j+1} \cdots p_{j+k-1} q_{j+k} + p_{j-1} p_{j+1} \cdots p_{j+k-2} q_{j+k-1} \\
&\quad + \cdots + p_{i+1} \cdots p_{j-1} p_{j+1} \cdots p_{k+i} q_{k+i-1}) \\
&= (p_i - p_j) [(q_{n-k+i} p_{n-k+i+1} \cdots p_n p_1 \cdots p_{i-1} - p_{j+1} \cdots p_{j+k-1} q_{j+k}) \\
&\quad + (q_{n-k+i+1} p_{n-k+i+2} \cdots p_n p_1 \cdots p_{i-1} p_{i+1} \\
&\quad \quad - p_{j-1} p_{j+1} \cdots p_{j+k-2} q_{j+k-1}) \\
&\quad + \cdots \\
&\quad + (q_{n-k+j-1} p_{n-k+j} \cdots p_n p_1 \cdots p_{i-1} p_{i+1} \cdots p_{j-1} - \\
&\quad \quad p_{i+1} \cdots p_{j-1} p_{j+1} \cdots p_{k+i} q_{k+i-1})].
\end{aligned}$$

In the square brackets of the above equation is a sum of the differences of two terms. Each term is a product of $k - 1$ component reliabilities and one component unreliability. The minuends include components starting from component l ($l = i - 1, i + 1, i + 2, \dots, j - 1$) counterclockwise (i is excluded), and the subtrahends include components starting from component m ($m = j + 1, j - 1, j - 2, \dots, i + 1$) clockwise (j is excluded). Because $n \leq 2k + 1$ and $j > i$, the ends of the corresponding two strings of k components clockwise and counterclockwise must overlap.

If we have the condition,

$$(p_i - p_j)(p_{i-1} - p_{j+1}) > 0 \quad \text{for all } j > i$$

satisfied, all the difference terms in the bracket will be positive or negative depending upon if $(p_i - p_j)$ is positive or negative, respectively, i.e., Δ will always be non-negative. This proves that Equation (5.6) is a necessary condition for the optimal design of a circular consecutive- k -out-of- n :G system with $n \leq 2k + 1$. (Q.E.D.)

Theorem 4 *For a circular consecutive- k -out-of- n :G system with $n \leq 2k + 1$, the only configuration satisfying the necessary condition in Theorem 3 is*

$$C_n = (1, 3, 5, 7, \dots, 8, 6, 4, 2, 1). \quad (5.7)$$

Thus, C_n is the invariant optimal design of a circular consecutive- k -out-of- n :G system with $n \leq 2k + 1$.

Proof of Theorem 4

To satisfy the necessary condition specified in Theorem 3, component 1 has to be adjacent to component 3 and 2. If not, say, 1 is adjacent to some i , $i > 3$. Let j be

the item following 2 in the sequence $1, i, \dots, 2, j$. This sequence violates the condition specified in Theorem 3, since $p_1 < p_j$ and $p_i > p_2$. Similarly, we can show that 2 must be adjacent to 1 and 4, 3 must be adjacent to 1 and 5, and so on. In essence, C_n is the only configuration satisfying the necessary condition in Theorem 3. Thus, it is the invariant optimal design of a circular consecutive- k -out-of- n :G system with $n \leq 2k + 1$. (Q.E.D.)

Theorem 5 *There does not exist any invariant optimal configuration for a linear consecutive- k -out-of- n :G system when $2 \leq k < n/2$.*

Proof of Theorem 5

To prove Theorem 5, it suffices to exhibit different choices of component reliabilities that lead to different optimal configurations for a linear consecutive- k -out-of- n :G system with $1 < k < n/2$.

Choose value T such that $0 < T < 1$, let $p_1 = p_2 = \dots = p_{n-k-1} = 0$, let $p_{n-k} = p_{n-k+1} = T$, and let $p_{n-k+2} = p_{n-k+3} = \dots = p_n = 1$. It is obvious that for this choice of component reliabilities — $k - 1$ perfect components, two equivalent imperfect components, and $n - k - 1$ failed components — the optimal configurations are:

$$(\underbrace{0, \dots, 0}_i, T, \underbrace{1, \dots, 1}_{k-1}, T, \underbrace{0, \dots, 0}_{n-i-k-1}), \quad (5.8)$$

where $i = 0, 1, \dots, n - k - 1$. The implication is that if an invariant optimal configuration exists, it must place the $k + 1$ best components together with the two worst components among these $k + 1$ best components at the two ends of the string.

Now choose S and T such that $0 < S < T < 1$, let $p_1 = p_2 = \dots = p_{n-k-1} = S$, let $p_{n-k} = p_{n-k+1} = T$, and let $p_{n-k+2} = p_{n-k+3} = \dots = p_n = 1$. From the above arguments, we know that the $k+1$ best components must be put together as $(T, 1, \dots, 1, T)$. We prove that the string has to be put at one end of the linear system to attain maximum system reliability, i.e.,

$$s1 = (T, \underbrace{1, \dots, 1}_{k-1}, T, \underbrace{S, \dots, S}_{n-k-1}). \quad (5.9)$$

The system's reliability of such a configuration, R_{s1} , is computed by decomposing on the two components with reliability T . This pair of components may have four states, $\{(0,0), (0,1), (1,0), (1,1)\}$, therefore,

$$R_{s1} = (1-T)^2 R(n-k-1) + 2T(1-T) + T^2, \quad (5.10)$$

where $R(n-k-1)$ is the reliability of a linear consecutive- k -out-of- $(n-k-1):G$ system with i.i.d. component reliability S .

Consider another configuration with i components of reliability S on the left of the $k+1$ best component string, i.e.,

$$s2 = (\underbrace{S, \dots, S}_i, T, \underbrace{1, \dots, 1}_{k-1}, T, \underbrace{S, \dots, S}_{n-i-k-1}), \quad (5.11)$$

where $i = 1, 2, \dots, n-k-2$. The system's reliability of such configurations, R_{s2} , is computed in a similar way,

$$R_{s2} = (1-T)^2 [R(i) + R(n-i-k-1) - R(i)R(n-i-k-1)] + 2T(1-T) + T^2, \quad (5.12)$$

where $R(i)$ is the reliability of a linear consecutive- k -out-of- $i:G$ system with i.i.d. component reliability S and $R(n-i-k-1)$ is the reliability of a linear consecutive- k -out-of- $(n-i-k-1):G$ system with i.i.d. component reliability S .

R_{s1} is greater than R_{s2} because $R(n-k-1)$ is greater than $R(i) + R(n-i-k-1) - R(i)R(n-i-k-1)$. $R(n-k-1)$ is the reliability of a G system with $n-k-1$ i.i.d. components of equal reliability S , and $R(i) + R(n-i-k-1) - R(i)R(n-i-k-1)$ is the sum of two independent non-null subsystem reliabilities and the total number of i.i.d. components of reliability S considered in these two subsystems is also $n-k-1$.

Thus, if an invariant optimal configuration exists, it must have the best $k+1$ components arranged as $(T, 1, \dots, 1, T)$ and put to one end of the linear system.

Finally, to obtain a contradiction, choose numbers S and T such that $0 < S < T < 0.5$. let $p_1 = p_2 = \dots = p_{n-k-1} = S$. let $p_{n-k} = p_{n-k-1} = \dots = p_n = T$. The previous computations imply that if an invariant optimal configuration exists, the configuration,

$$s1 = (\underbrace{T, \dots, T}_{k+1}, \underbrace{S, \dots, S}_{n-k-1}) \quad (5.13)$$

must be optimal for this particular choice of component reliabilities. This is not the case, in fact, we can prove that another configuration,

$$s2 = (S, \underbrace{T, \dots, T}_{k+1}, \underbrace{S, \dots, S}_{n-k-2}) \quad (5.14)$$

is strictly better. We will use $R(p_1, \dots, p_n)$ to represent the reliability of a linear consecutive- k -out-of- n :G system with component reliabilities arranged as p_1, p_2, \dots, p_n .

$$R_{s2} = R(\underbrace{T, \dots, T}_{k+1}, \underbrace{S, \dots, S}_{n-k-1}) + Q(\underbrace{T, S, \dots, S}_{n-k-1})ST^{k-1}(1-T). \quad (5.15)$$

$$R_{s1} = R(\underbrace{T, \dots, T}_{k+1}, \underbrace{S, \dots, S}_{n-1}) + \begin{cases} Q(\underbrace{T, \dots, T}_{k+1}, S, \dots, S)(1-S)S^k, & n > 2k+1 \\ Q(\underbrace{T, \dots, T}_{n-k-1})(1-T)S^k, & n = 2k+1 \\ (1-T)T^{2k+1-n}S^{n-k-1}, & k < n < 2k+1 \end{cases} \quad (5.16)$$

We say that R_{s2} is greater than R_{s1} because of the following inequalities.

$$\begin{aligned} Q(\underbrace{T, S, \dots, S}_{n-k-1}) &> Q(\underbrace{T, \dots, T}_{k+1}, S, \dots, S), & \text{when } n \geq 2k+1 \\ Q(\underbrace{T, S, \dots, S}_{n-k-1}) &= 1, & \text{when } n < 2k+1 \\ ST^{k-1}(1-T) &> (1-T)T^{2k+1-n}S^{n-k-1}, & \text{when } n < 2k+1 \\ ST^{k-1}(1-T) &> (1-T)S^k, & \text{when } n = 2k+1 \\ ST^{k-1}(1-T) &> (1-S)S^k, & \text{when } n > 2k+1. \end{aligned}$$

The last inequality is true from the following inequalities:

$$\begin{aligned} 0 &< S < T < 0.5 \\ T(1-T) &> S(1-S) \\ T^{k-1}(1-T) &> S^{k-1}(1-S) \\ ST^{k-1}(1-T) &> (1-S)S^k. \end{aligned}$$

Therefore, there are cases when R_{s2} is larger than R_{s1} , i.e., there does not exist an invariant optimal configuration for a linear consecutive- k -out-of- n :G system when $1 < k < n/2$. (Q.E.D.)

Theorem 6 *There does not exist any invariant optimal configuration for a circular consecutive- k -out-of- n :F system when $3 \leq k < n - 2$.*

Proof of Theorem 6

To prove Theorem 6, it suffices to exhibit different choices of component reliabilities that lead to different optimal configurations for a linear consecutive- k -out-of- n :F system with $3 \leq k < n - 2$.

First, consider the case when $k \neq n - 3$. Choose $p_n = 1$, and then the circular consecutive- k -out-of- n :F system is equivalent to a linear consecutive- k -out-of- $(n - 1)$:F system with component reliabilities p_1, p_2, \dots, p_{n-1} and $k < (n - 1) - 2$. According to Theorem 1 of Malon [57], the linear consecutive- k -out-of- $(n - 1)$:F system does not have an invariant optimal configuration. As a result, the circular consecutive- k -out-of- n :F system with $p_n = 1$ does not have an invariant optimal configuration. Thus, a general circular consecutive- k -out-of- n :F system with $3 \leq k < n - 3$ does not either.

Now consider the case when $k = n - 3$. If $p_n = 1$, then the rest $n - 1$ components should be ordered following the invariant optimal configuration of a linear consecutive- $(n - 3)$ -out-of- $(n - 1)$:F system as specified in Malon [57]. Choose S and T such that $0 < S < T < 1$, let $p_1 = p_2 = S$, and let $p_3 = p_4 = \dots = p_{n-1} = T$. Then the optimal configuration is supposed to be

$$(1, S, \underbrace{T, \dots, T}_{n-3}, S, 1), \quad (5.17)$$

i.e., the two least reliable components should be put adjacent to the most reliable component. However, we find the following choice of component reliabilities contradicts the above configuration.

Choose T such that $0 < T < 1$, let $p_1 = p_2 = 0$, and let $p_3 = p_4 = \dots = p_n = T$. If the above configuration is optimal we need to arrange the components in the order.

$$s1 = (T, 0, \underbrace{T, \dots, T}_{n-3}, 0, T) \quad (5.18)$$

and the system's reliability is (decomposing on a component with 0 reliability with the formula in Antonopoulou and Papastavridis [4]):

$$R_{s1} = R_C(\underbrace{T, \dots, T}_{n-1}, 0) - T^2(1-T)^{k-1} - (k-2)T^2(1-T)^{k-2}. \quad (5.19)$$

However, the following configuration.

$$s2 = (T, T, 0, \underbrace{T, \dots, T}_{n-4}, 0, T), \quad (5.20)$$

has system reliability (calculated in a similar way):

$$R_{s2} = R_C(\underbrace{T, \dots, T}_{n-1}, 0) - T^2(1-T)^{k-1} - (k-3)T^2(1-T)^{k-2}, \quad (5.21)$$

which is larger than R_{s1} . (Q.E.D.)

Theorem 7 *There does not exist any invariant optimal configuration for a circular consecutive- k -out-of- n :G system when $2 \leq k < (n-1)/2$.*

Proof of Theorem 7

To prove Theorem 7, it suffices to exhibit different choices of component reliabilities that lead to different optimal configurations for a linear consecutive- k -out-of- n :G system with $2 \leq k < (n-1)/2$.

Choose $p_1 = 0$. Then, the circular consecutive- k -out-of- n :G system is equivalent to a linear consecutive- k -out-of- $(n - 1)$:G system with component reliabilities p_2, p_3, \dots, p_n and $k < (n - 1)/2$. According to Theorem 5 above, the linear consecutive- k -out-of- $(n - 1)$:G system does not have an invariant optimal configuration. As a result, the circular consecutive- k -out-of- n :G system with $p_1 = 0$ does not have an invariant optimal configuration, and thus, not a general circular consecutive- k -out-of- n :G system with $k < (n - 1)/2$. (Q.E.D.)

With the theorems developed in this section, the theory of the optimal design of the linear and circular consecutive- k -out-of- n systems are complete. They are summarized in the following two tables.

Variant Optimal System Designs

This section is devoted to the designs of consecutive- k -out-of- n systems where invariant optimal designs do not exist. This section presents a heuristic method and a randomization method to find at least sub-optimal designs of consecutive- k -out-of- n systems. A binary search method is also proposed to find optimal designs of a linear consecutive- k -out-of- n system with $n \leq 2k$.

Heuristic method

The heuristic is that a position with a higher Birnbaum component reliability importance should be assigned a component with larger reliability, or in other words, the reliability pattern matches the component reliability importance pattern. To test the goodness of the heuristic, an exhaustive search was used to find optimal system

Table 5.1: Invariant optimal designs of linear consecutive- k -out-of- n systems

k	F System	G System
$k = 1$	(any arrangement)	(any arrangement)
$k = 2$	$(1, n, 3, n - 2, \dots, n - 3, 4, n - 1, 2)$ Derman, Lieberman, and Ross	(Does not exist) ^a
$2 < k < \frac{n}{2}$	(Does not exist)	
$\frac{n}{2} \leq k < n - 2$	Malon	$(1, 3, 5, \dots, 2(n - k) - 1, (\text{any arrangement}), 2(n - k), \dots, 6, 4, 2)$ Kuo, Zhang, and Zuo
$k = n - 2$	$(1, 4, (\text{any arrangement}), 3, 2)$ Malon	
$k = n - 1$	$(1, (\text{any arrangement}), 2)$ Malon	
$k = n$	(any arrangement)	(any arrangement)

^aResult developed in this dissertation.

Table 5.2: Invariant optimal designs of circular consecutive- k -out-of- n systems

k	F System	G System
$k = 1$	(any arrangement)	(any arrangement)
$k = 2$	$(1, n - 1, 3, n - 3, \dots, n - 4, 4, n - 2, 2, n, 1)$ Hwang	(Does not exist) ^a
$2 < k < \frac{n-1}{2}$	(Does not exist) ^a	
$\frac{n-1}{2} \leq k < n - 2$		
$k = n - 2$	$(1, n - 1, 3, n - 3, \dots, n - 4, 4, n - 2, 2, n, 1)^a$	
$k = n - 1$	(any arrangement) Kuo, Zhang, and Zuo	(any arrangement) Kuo, Zhang, and Zuo
$k = n$	(any arrangement)	(any arrangement)

^aResult developed in this dissertation.

designs.

Assume that there are n components with their reliability values known. Initially each of the n positions is assigned a component (this assignment is an initial design). Then the Birnbaum importance of each component is calculated. If a position, say i , has a more reliable component but not a higher importance than another position, say j , then these two components exchange their positions. This process continues until the importance pattern matches the reliability pattern or no interchange of any two components improves system reliability.

There are two problems in implementing the heuristic method. One is how to assign the initial design and the other is how to improve the design if it does not satisfy the heuristic condition. A few initial designs and criteria to improve the design are compared in order to select a good initial design and a good way to improve the current solution,

Initial Design 1: $(1, n-1, 3, n-3, \dots, n-2, 4, n, 2)$, the importance pattern of a consecutive-2-out-of- n system with i.i.d. components.

Initial Design 2: $(1, 3, 5, \dots, n, \dots, 6, 4, 2)$, a pattern found to be optimal in many cases with exhaustive search.

Initial Design 3: $(1, n, 3, n-2, \dots, n-3, 4, n-1, 2)$, the optimal pattern for a consecutive-2-out-of- $n:F$ system.

Initial Design 4: $(1, 2, 3, 4, \dots, n)$, a naturally ordered pattern.

Heuristic 1: Starting from the least reliable component, the reliability importance of this component is compared with the reliability importance of the next more reliable component. If the importance of the less reliable component is larger than that of the more reliable component, exchange these two components. If the system's reliability is improved by this exchange, the exchange is kept. Otherwise, the exchange is abolished, and the next more reliable component is considered. The process continues until either the reliability pattern matches the importance pattern or no more exchange can improve the system's reliability.

Heuristic 2: Starting from the least reliable component, its importance is compared with the importances of all the components with higher reliabilities. If this component is not the least important one among these components, its position is exchanged with the one least important. If the interchange of the two components improves system reliability, the interchange is kept, otherwise the interchange is abolished and the next more reliable component is considered. The process continues until the component importance pattern matches the reliability pattern or no more interchanges can improve system reliability.

Test on regularly generated component reliabilities To test the above scenarios and select a better initial solution and a good way to improve system reliability through design, we use the following data:

$n = 7$, G system

$A = 0.1 \times i$, for $i = 1, \dots, 9$

$$B = \frac{1 - A}{30} \times j, \quad \text{for } j = 1, \dots, 29$$

$$p_l = A + B \times (l - 1), \quad \text{for } l = 1, 2, \dots, 7.$$

Two hundred and sixty-one sets of component reliabilities were created. For each of the 261 sets of components, the optimal system design was obtained with an exhaustive search and a best system design was obtained using one of the above two heuristics with one of the above four initial designs. All system reliabilities obtained were standardized by dividing them by the corresponding real optimal system reliability for that set of components. The results are tabulated in Table 5.3 and Table 5.4.

In Tables 5.3 and 5.4 *Mean Rel.* represents the mean standard reliability for the 261 sets of components; *Std. Dev.* the standard deviation of the 261 standardized reliabilities; *Min. (Max.)* the minimum (maximum) standard system reliability among the 261 standardized reliabilities; and *Sum* the total of the 261 standardized system reliabilities. Thus, if the standardized system reliability is 1, this system reliability is optimal, and if the standardized system reliability is 0.95, then the system reliability given is only 95% of the real optimal system reliability.

From Table 5.3, it is clear that the closest results were obtained with initial design 2. The average reliability provided by this heuristic is 99.98% of the optimal design for both $k = 2$ and $k = 3$. Even the worst ones are 99.83% ($k = 2$) and 99.30% ($k = 3$) of the optimal system reliabilities, respectively. The worst average result was obtained with initial pattern 4, which provides designs with system reliability about 99.60% ($k = 2$) and 98.88% ($k = 3$) of the optimal system reliability, respectively.

From Tables 5.3 and 5.4 we find that Heuristic 1 is slightly better than Heuristic

Table 5.3: Comparison of initial solutions on optimal system design with Heuristic 1

k	Item	Mean Rel.	Std. Dev.	Min.	Max.	Sum
$k = 2$	Optimal	1.0000	0.0000	1.0000	1.0000	261.0000
	Initial 1	0.9974	0.0066	0.9583	1.0000	260.3331
	Initial 2	0.9998	0.0003	0.9983	1.0000	260.9452
	Initial 3	0.9971	0.0054	0.9744	1.0000	260.2345
	Initial 4	0.9960	0.0085	0.9583	1.0000	259.9656
$k = 3$	Optimal	1.0000	0.0000	1.0000	1.0000	261.0000
	Initial 2	0.9998	0.0008	0.9930	1.0000	260.9381
	Initial 4	0.9888	0.0222	0.9134	1.0000	258.0748

Table 5.4: Comparison of initial solutions on optimal system design with Heuristic 2

k	Item	Mean Rel.	Std. Dev.	Min.	Max.	Sum
$k = 2$	Optimal	1.0000	0.0000	1.0000	1.0000	261.0000
	Initial 1	0.9968	0.0064	0.9686	1.0000	260.1520
$k = 3$	Optimal	1.0000	0.0000	1.0000	1.0000	261.0000
	Initial 2	0.9998	0.0008	0.9930	1.0000	260.9371

2. For $k = 2$ with initial design 1, the system's reliability obtained with Heuristic 1 is 99.74% of the optimal system reliability on the average, while the system's reliability obtained with Heuristic 2 is 99.68% of the optimal system reliability on the average. As a result, we selected Heuristic 1 and initial design 2 as our heuristic method. Further discussions are provided below.

Test on component sets randomly generated In order to further investigate the heuristic method and measure its performance for both F and G systems, 100 sets of components of size n ($n = 7$ or 8) were generated with a random number generator. For example, one set of components of size 7 have reliabilities 0.102167, 0.152044, 0.152653, 0.171663, 0.393335, 0.595365, and 0.759706. These components were numbered in a non-decreasing order of their reliabilities. Thus, for the example set of components we have

Comp no.	1	2	3	4	5	6	7
Comp rel.	0.102167	0.152044	0.152653	0.171663	0.393335	0.595365	0.759706

For each of the 100 sets of randomly generated components, the heuristic was applied to find the sub-optimal solutions, and then an exhaustive search was used to find the optimal solutions. The results are tabulated in Table 5.5. From the table we see that the heuristic provides solutions with 96.9% of the optimal system reliability on the average for an F system with $k = 3$ and $n = 7$. For the G system or larger n values, the solutions are better. For example, for $n = 8$ and $k = 3$ the solution is 99.99% of the real optimal on the average.

Since most commercial products have a relatively high component reliability, we assumed that the component reliabilities were in the range of (0.8, 0.99). One

Table 5.5: Test results on 100 sets of randomly generated component reliabilities in range of (0.1)

n		$n = 7$		$n = 8$	
system	Item	$k = 2$	$k = 3$	$k = 2$	$k = 3$
F	Mean Rel.		0.969		0.9999
	Min.		0.505		0.9983
	Max.		1.000		1.0000
G	Mean Rel.	0.998	0.999	0.9986	0.9987
	Min.	0.977	0.988	0.9775	0.9887
	Max.	1.000	1.000	1.0000	1.0000

hundred sets of component reliabilities of sizes 7 and 8 in this range were generated. The optimal solutions and heuristic solutions were found. They are tabulated in Table 5.6. From the tabulated data, we can see the heuristic is very good. It gives a solution that is over 99% of the optimal solution on the average.

Table 5.6: Test results on 100 sets of randomly generated component reliabilities in range of (0.8,0.99)

n		$n = 7$		$n = 8$	
system	Item	$k = 2$	$k = 3$	$k = 2$	$k = 3$
F	Mean Rel.		1.000		1.000
	Min.		0.998		0.998
	Max.		1.000		1.000
G	Mean Rel.	1.000	1.000	1.000	1.000
	Min.	1.000	0.998	1.000	0.999
	Max.	1.000	1.000	1.000	1.000

Randomization method

The exhaustive search method enumerates all possible system configurations and finds the configuration with maximum system reliability. For a linear consecutive- k -out-of- n system, the number of configurations to compare is $n!/2/k!/2$ considering symmetry and the necessary conditions of optimal system design. For a circular consecutive- k -out-of- n system, the number of configurations to compare is $(n-1)!$.

The methodology of the randomization method is to compare only a limited number of possible configurations instead of all, and choose the best one among these configurations. The closeness of the solution obtained with this method depends upon the number of configurations compared and the quality of the random number generator. The larger the number of the configurations compared, the better the solution obtained, and also the longer the computation time.

The steps of implementation of this method are:

1. Arrange components in ascending order of their reliabilities.
2. Generate a random permutation of integer numbers from 1 to n .
3. Rearrange the left k numbers in ascending order and the right k numbers in descending order for a linear consecutive- k -out-of- n system.
4. Compute the system's reliability with the configuration in previous steps.
5. If enough permutations have been generated, then stop; otherwise, go back to step 1.

Binary search method

For the optimal system design of a general linear consecutive- k -out-of- n system, we have discussed a heuristic method and a randomization method obtaining at least suboptimal solutions. These two methods may be used for any consecutive- k -out-of- n system, but the shortcoming is that it cannot guarantee optimal solutions.

In order to obtain the real optimal solution of a general consecutive- k -out-of- n system an enumeration method must be used, unless there exists an invariant optimal design. The complexity of an enumeration method is $n!$. In the following, a binary search method is proposed to find the optimal solution for the linear consecutive- k -out-of- n :F systems with $n/2 \leq k \leq n$.

The rational for the method is based upon the necessary conditions for the optimal design of a linear consecutive- k -out-of- n system, that is, the leftmost k components should be ordered in a non-decreasing order of their reliability, the rightmost k components should be ordered in a non-increasing order of their reliability, and the overlapped components may be ordered in any order.

For a consecutive- k -out-of- n :F system with $k = n/2$ and n is even, apparently at least one of positions k and $k + 1$ should have the most reliable component. From this position leftward and rightward, the remaining components should be ordered in a non-increasing order of component reliabilities.

For a consecutive- k -out-of- n system with $k > n/2$, apparently positions $n - k - 1$ through k should have the $2k - n$ most reliable components arranged in any order. Then, the remaining components are assigned to the positions on the left and on the right in a non-increasing order of their reliabilities from the central $2k - n$ positions.

The complexity of this method is $C(2(n-k), n-k)$ (the number of combinations of $2(n-k)$ taken $n-k$ at a time) for $n < 2k$, and $C(n-1, k-1)$ for $n = 2k$.

Summary

This study developed a heuristic method for optimal design of a general linear consecutive- k -out-of- n system. The heuristic provides very close-to-optimal solutions. A binary search method was presented to find optimal designs of a linear consecutive- k -out-of- n system with $n \leq 2k$. Invariant optimal designs for linear and circular consecutive- k -out-of- n systems were identified and the theories on optimal design of consecutive- k -out-of- n systems were completed.

CHAPTER 6. CASE STUDIES

Two examples of consecutive- k -out-of- n :G systems are presented in this chapter to illustrate the applications of the results of optimal system designs reported in Chapter 5. One is a linear consecutive- k -out-of- n :G system and the other is a circular consecutive- k -out-of- n :G system. Though reliability is the measure of performance that has been discussed in this dissertation, the same concept and methods can be used for similar measures of performance like system availability.

A Sea Port

An example of a linear consecutive- k -out-of- n :G system is a sea port with n berths of standard size for ships to dock. The berths are numbered consecutively from 1 to n . A regular ship entering the port needs one berth for cargo loading and unloading. However, if a large ship enters the port, k ($k \leq n$) consecutive berths are needed because of its size and loading and unloading facilities involved. In this case, if there are at least k consecutive berths available in the port, the ship can enter the port for service. Otherwise, the ship cannot come into the port and must wait in the sea until there are k consecutive berths available. Much cost is involved for the ship to wait out of the port. The problems of interest are how to evaluate the system availability (the probability that such a large ship can enter the port without delay)

and how to operate the port to maximize such a probability.

Let us assume that the number of berths in the port is seven, a large ship takes four consecutive berths, and on the average 3.5 berths are in use at any time (i.e., the port's utilization factor is 50%). If we further assume that the utilization factors of the berths are identical, then we have a linear consecutive-3-out-of-7:G system with the following system parameters:

$$n = 7$$

$$k = 4$$

$$q = \frac{3.5}{7} = 0.5$$

$$p = 1 - q = 0.5.$$

where q is the berth utilization factor and p is the berth availability.

Using Formula (4.11), the system availability is calculated below.

$$\begin{aligned} R_G(7; 4) &= \sum_{i=1}^4 p^4 - \sum_{i=1}^3 p^5 = p^4(4 - 3p) \\ &= 0.5^4 \times (4 - 3 \times 0.5) = 0.15625. \end{aligned}$$

In other words, if the port's utilization factor is 50% and the port is managed such that the seven berths are equally utilized, then the system's availability is 0.15625, i.e., a large ship has a probability of about 0.16 to enter the port without delay. The Birnbaum importances of all the berths are calculated with Formula (3.49) and tabulated in Table 6.1. From the table we can see that berth 4 is the most important one and the berths become more important as their positions are closer to berth 4. This means that the berths in the middle are the ones whose availabilities should

be increased first in order to increase the system's availability the most. Another phenomenon observed from the table is that berth importance is symmetric to the middle position of the line of berths.

Table 6.1: Berth availability importances when the berths are i.i.d.

Berth No.	Availability	Importance	Importance Rank
1	0.5	0.0625	4
2	0.5	0.1250	3
3	0.5	0.1875	2
4	0.5	0.3125	1
5	0.5	0.1875	2
6	0.5	0.1250	3
7	0.5	0.0625	4

Now consider the case when the berths are not equally utilized, e.g., there are seven different utilization factors to be assigned to the seven berths such that the average number of berths in use is still 3.5. A set of berth availabilities satisfying such a condition is presented in ascending order in Table 6.2.

Table 6.2: A set of berth availabilities

i	1	2	3	4	5	6	7
$p[i]$	0.35	0.4	0.45	0.5	0.55	0.6	0.65
$u[i]$	0.65	0.6	0.55	0.5	0.45	0.4	0.35

With the berth availabilities given in Table 6.2, there is a problem of assigning them to different berths such that the system's availability is maximized. According to Kuo, Zhang, and Zuo [49], there exist invariant optimal configurations for the consecutive-4-out-of-7:G system, since $k = 4 > \frac{n}{2} = 3.5$. With the assumption that

the berth availabilities are already numbered in ascending order of their reliabilities as in Table 6.2, the invariant optimal arrangements for a linear consecutive- k -out-of- n :G system with $k > n/2$ are to put the odd numbered components starting from one end in ascending order, to put the even numbered components starting from the other end in ascending order, and to put the components remaining to the $2k - n$ positions in the middle in any order. Applying this general rule to this example, we have the following invariant optimal configuration.

$$v = (1, 3, 5, 7, 6, 4, 2). \quad (6.1)$$

Therefore,

$$p_1 = 0.35$$

$$p_2 = 0.45$$

$$p_3 = 0.55$$

$$p_4 = 0.65$$

$$p_5 = 0.6$$

$$p_6 = 0.5$$

$$p_7 = 0.4.$$

The system's availability with the optimal configuration in Equation (6.1) is 0.213135. We can see that the new arrangement guarantees that the berths in the middle have larger availabilities. By managing the port this way, the management has increased the port's availability from 0.15625 to 0.213135, a 36.4% increase without decreasing the port's utilization factor. The berth availability importances with the optimal

arrangement in Equation (6.1) are tabulated in Table 6.3. It is still true that the berths in the middle have larger availability importances. Thus, it will benefit the management the most if the availabilities of the berths in the middle are further increased.

Table 6.3: Berth availability importances with optimal arrangement

Berth No.	Availability	Importance	Importance Rank
1	0.35	0.0643503	7
2	0.45	0.1573001	5
3	0.55	0.2457000	3
4	0.65	0.3279000	1
5	0.6	0.2613812	2
6	0.5	0.1881750	4
7	0.4	0.0877500	6

In summary, for the cases where $n \leq 2k$, our suggestion to the management of the port is to assign regular ships to the berths at two ends and leave the berths in the middle less utilized as much as possible. The berths in the middle are saved in case a large ship that needs four consecutive berths arrives. In this way, the system availability for a large ship is maximized.

If a large ship needs two consecutive berths available, instead of four, to enter the port, then there does not exist any invariant optimal arrangement of berth availabilities, since $k = 2 < \frac{n}{2} = 3.5$ according to Theorem 5 in Chapter 5. The optimal arrangement depends on the values of the berth availabilities. The only way to determine the optimal solution for a set of berth availabilities in this case is to enumerate all possible arrangements that satisfy the necessary conditions of optimal solution [49]. The enumeration method is used for this example and the optimal

solution found is

$$v = (1, 3, 4, 5, 7, 6, 2),$$

and the optimal system availability is 0.77356875. Table 6.4 lists the top five configurations and their corresponding system availabilities.

Table 6.4: Top five configurations from data in Table 6.2

Rank	Configuration	System Availability
1	(1,3,4,5,7,6,2)	0.773568750
2	(1,3,4,6,7,5,2)	0.773507500
3	(1,3,5,7,6,4,2)	0.773229375
4	(1,4,3,5,7,6,2)	0.772956250
5	(1,3,6,7,5,4,2)	0.772901250

If the system size is very large, it is impossible to use the enumeration method. The heuristic method discussed in Chapter 5 may be used to obtain sub-optimal solutions. Applying the heuristic method to this example, we have obtained a configuration of (1, 3, 5, 7, 6, 4, 2) and the system availability is 0.773229. This result is very close to the real optimal solution (99.956% of the real optimal solution). It is the third best solution (see Table 6.4).

The randomization method may also be used to obtain a suboptimal solution. The closeness of the solution obtained to the optimal solution depends on the number of random configurations evaluated. With m denoting the number of random configurations generated in this example, three cases were considered, $m = 10$, $m = 100$, and $m = 1000$. For each case, 10 runs were made. The best configuration and system availability and the average system availability for each case were obtained and are listed in Table 6.5. We see that the best solution of 10 runs in the cases of both

$m = 100$ and $m = 1000$ are the real optimal. However, on the average, the three cases provides configurations with 99.33%, 99.92%, and 99.99% of the real optimal system availability.

Table 6.5: Results obtained with the randomization method

m	Average Availability & Its % of Real Optimal	Best Availability	Best Pattern
10	0.768376 99.93	0.772901250	(1,3,6,7,5,4,2)
100	0.772938 99.92	0.773568750	(1,3,4,5,7,6,2)
1000	0.773489 99.99	0.773568750	(1,3,4,5,7,6,2)

Photograph Techniques in a Nuclear Accelerator

To analyze the acceleration activities that happen in a nuclear accelerator, high speed cameras are used to take pictures of the action. Because of the speed of the action and the cost involved in implementing such an experiment, the photographing system must be very reliable and accurate. To ensure the proper functioning and the quality of the pictures taken, a circular consecutive- k -out-of- n :G system of cameras is used. A set of n cameras are installed around the accelerator and if and only if at least k consecutive cameras work properly can the photographing system work properly. The problems of interest are the calculation of the reliability of the photographing system and the optimal design of the system if cameras of different reliabilities are used.

Assume the following parameters for the circular consecutive- k -out-of- n :G sys-

tem:

$$n = 7$$

$$k = 3$$

$$p = 0.8$$

$$q = 0.2.$$

Then the system's reliability is calculated below with Formula (4.15):

$$\begin{aligned} R_{CG}(7;3) &= \sum_{i=1}^n p^k - \sum_{i=1}^n p^{k+1} + p^n \\ &= 7 \times 0.8^3 - 7 \times 0.8^4 + 0.8^7 \\ &\approx 0.9265. \end{aligned}$$

The component importances are all equal to 0.134144. The sensitivities of the system reliability to all component reliabilities are identical. The improvement of any single component has an equal effect on the system's reliability.

Table 6.6: A set of camera reliabilities

Number	1	2	3	4	5	6	7
Reliability	0.65	0.7	0.75	0.8	0.85	0.9	0.95

Assume that there are seven numbered cameras with different reliabilities, as tabulated in Table 6.6. As shown in the table, camera 1 has the lowest reliability and camera 7 has the highest reliability. The way these cameras are arranged around the accelerator will make a difference in a system's reliability. According to Kuo, Zhang,

and Zuo [49], the invariant optimal configuration for a circular consecutive- k -out-of- n :G with $k \geq (n - 1)/2$ is

$$w = (1, 3, 5, \dots, 2(n - k) - 1, (\text{any arrangement}), 2(n - k), \dots, 6, 4, 2, 1). \quad (6.2)$$

For the example in this study, $k = 3 \geq (n - 1)/2 = 3$, therefore, the optimal arrangement of the cameras is

$$w = (1, 3, 5, 7, 6, 4, 2, 1). \quad (6.3)$$

Therefore,

$$p_1 = 0.65$$

$$p_2 = 0.75$$

$$p_3 = 0.85$$

$$p_4 = 0.95$$

$$p_5 = 0.90$$

$$p_6 = 0.80$$

$$p_7 = 0.70.$$

Figure 6.1 shows the optimal arrangement of cameras around the accelerator. The system's reliability with the optimal design in Equation (6.3) is 0.9488. By arranging the cameras this way, the system's reliability is maximized.

The camera reliability importances with the optimal design are tabulated in Table 6.7. From Table 6.7 we see that a camera with higher reliability also has a higher importance. This reminds us to use the same heuristic idea to find sub-optimal solutions when there does not exist invariant optimal solutions.

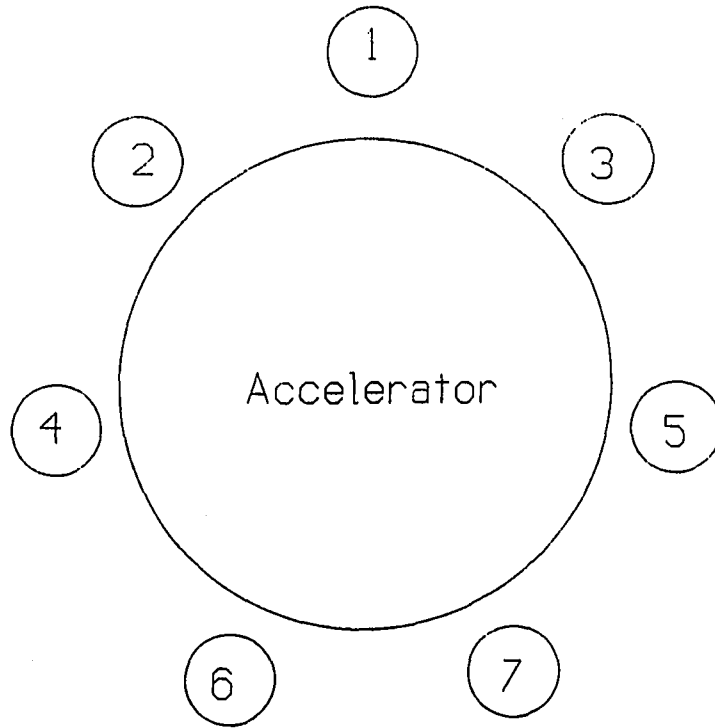


Figure 6.1: The optimal arrangement of cameras

Table 6.7: Camera reliability importances with the optimal design

Position	Reliability	Importance
1	0.65	0.0508600
2	0.75	0.0557120
3	0.85	0.0765400
4	0.95	0.1105621
5	0.90	0.0901975
6	0.80	0.0637691
7	0.70	0.0545575

Table 6.8: The top five configurations from data in Table 6.6

Rank	Reliability	Configuration
1	0.995082937	(1 2 3 4 6 7 5 1)
2	0.995068562	(1 2 5 7 6 4 3 1)
3	0.995041062	(1 2 6 7 5 4 3 1)
4	0.995016687	(1 2 4 3 6 7 5 1)
5	0.995007312	(1 2 4 3 5 7 6 1)

To show the application of the heuristic method to a circular consecutive- k -out-of- n :G system where invariant optimal configurations do not exist, assume $k = 2$. According to Theorem 7 in Chapter 5, no invariant optimal configurations exist because $k = 2 < (n - 1)/2 = 3$. The actual optimal configuration is component reliability dependent.

Using the component reliability data in Table 6.6, the optimal solution is obtained with the enumeration method for comparison with the sub-optimal solutions obtained with other methods. The optimal configuration is (1,2,3,4,6,7,5) and the corresponding system reliability is 0.995083. The top five configurations are tabulated in Table 6.8.

The heuristic method described in Chapter 5 is used. The result is shown in Table 6.9. As shown in the table, the best configuration obtained is (1, 2, 5, 7, 6, 4, 3, 1) and the corresponding system's reliability is 0.995069. Compared with the real optimal solution, the heuristic provides a solution that is 99.9986% of the optimal solution in this case, practically no difference with the real optimal exists at all.

The randomization method may also be used to obtain sub-optimal solutions. For the case of the circular system, the method randomly generates configurations

Table 6.9: Heuristic result for the circular consecutive- k -out-of- n :G system

Pattern	Com. Rel.	Com. Imp.
1	0.65	0.008036
2	0.75	0.008130
3	0.85	0.014663
4	0.95	0.024477
5	0.90	0.019262
6	0.80	0.010384
7	0.70	0.010066

Table 6.10: Sub-optimal solution obtained with the randomization method

m	Average Reliability	Best Reliability	Best Pattern
10	0.994560	0.995041	(1 3 4 5 7 6 3)
100	0.995019	0.995069	(2 1 3 4 6 7 5)
700	0.995082	0.995083	(1 2 3 4 6 7 5)

and calculates the corresponding system reliability. The best solution of a certain number of configurations specified is used as a sub-optimal solution. The solutions obtained with this method are presented in Table 6.10. With m denoting the number of random configurations evaluated, three cases were considered, $m = 10$, $m = 100$, and $m = 700$. For each case, 10 runs were made. The best configuration, system reliability, and the average system reliability for each case are obtained and listed in Table 6.10. For the case of $m = 700$ the real optimal was reached. However, the average system reliability obtained in each case was 99.95%, 99.99%, and 100.00% of the real optimal reliability, respectively.

Summary

The examples discussed in this chapter provide applications of the system's reliability or availability evaluation and optimal system design of both linear and circular consecutive- k -out-of- n :G systems. With the theories developed in this dissertation and proper application (as shown in this chapter), the most benefit could be achieved. In the cases where there are no invariant optimal configurations, the heuristic method provides a quick way to obtain close-to-optimal solutions.

CHAPTER 7. SUMMARY AND CONCLUSIONS

This research studies two special systems: k -out-of- n systems (F & G) and consecutive- k -out-of- n systems (F & G). A k -out-of- n :F system fails if and only if at least k of its n components fail. A k -out-of- n :G system is good if and only if at least k of its n components are good. A consecutive- k -out-of- n :F system is a sequence of n ordered components such that the system works if and only if less than k consecutive components fail. A consecutive- k -out-of- n :G system consists of an ordered sequence of n components such that the system works if and only if at least k consecutive components in the system are good. The consecutive- k -out-of- n systems are further divided into linear systems and circular systems corresponding to the cases that the components are ordered along a line and a circle, respectively.

After the reliability evaluation of the k -out-of- n systems and the reliability evaluation and optimal design of the consecutive- k -out-of- n systems were reviewed, the properties of these systems were further investigated in this research. A set of special system reliability formulas were developed. The Sum of Disjoint Path method was used to develop the reliability formula for a linear consecutive- k -out-of- n :G system.

Next, this research concentrated on the optimal design of the consecutive- k -out-of- n systems. An arrangement of components is optimal if it maximizes the system's reliability. An optimal arrangement is invariant if it depends only on the ordering

of component reliabilities and not their actual values. All n and k combinations of linear and circular consecutive- k -out-of- n systems were studied. Theorems were developed to identify invariant optimal designs of some consecutive systems if they existed. Other theorems were provided to prove that there were no invariant optimal configurations for other consecutive systems. The complete theories for the optimal system design of the consecutive- k -out-of- n systems were accomplished.

For those systems where invariant optimal designs do not exist a heuristic method was provided to find approximate optimal solutions. The randomization method was used to compare performances of the heuristic method. Two examples of consecutive- k -out-of- n :G systems were provided to illustrate uses of the theoretical results developed in this research.

Research in the area of consecutive- k -out-of- n systems may be done further. Some special consecutive- k -out-of- n systems have been presented by other researchers. These special systems may be investigated more deeply. Most of current research has used single values of component and system reliabilities. Distributions of component reliabilities and system reliability may be assumed and investigated. The methods and ideas from the research of consecutive- k -out-of- n systems may be adapted to other special systems.

BIBLIOGRAPHY

- [1] Abraham, J. A. "An improved method for network reliability." *IEEE Transactions on Reliability*, R-28 (Apr 1979): 58-61.
- [2] Aggarwal, K. K., K. B. Misra, and J. S. Gupta. "A fast algorithm for reliability evaluation." *IEEE Transactions on Reliability*, R-24 (Apr 1975): 83-85.
- [3] Aggarwal, K. K. and Suresh Rai. "Reliability evaluation in computer communication networks." *IEEE Transactions on Reliability*, R-30 (Apr 1981): 32-35.
- [4] Antonopoulou, I. and S. Papastavridis. "Fast recursive algorithm to evaluate the reliability of a circular consecutive- k -out-of- n :F system." *IEEE Transactions on Reliability*, R-36 (Apr 1987): 83-84.
- [5] Barlow, R. E. and K. D. Heidtmann. "Computing k -out-of- n system reliability." *IEEE Transactions on Reliability*, R-33 (Oct 1984): 322-323.
- [6] Barlow, R. E. and F. Proschan. *Mathematical Theory of Reliability*. John Wiley & Sons, New York, 1967.
- [7] Barlow, R. E. and F. Proschan. *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart & Winston, New York, 1975.
- [8] Billinton, R. and R. N. Allan. *Reliability Evaluation of Engineering Systems: Concepts and Techniques*. Plenum Press, New York, 1983.
- [9] Birnbaum, Z. W. "On the importance of different components in a multi-component system." In *Multivariate Analysis—II*, ed. P. R. Krishnaiah, 581-592. Academic Press, New York, 1969.
- [10] Bollinger, R. C. "Direct computation for consecutive- k -out-of- n :F systems." *IEEE Transactions on Reliability*, R-31 (Dec 1982): 444-446.
- [11] Bollinger, R. C. "Reliability and runs of ones." *Mathematics Magazine*, 57 (Jan 1984): 34-37.

- [12] Bollinger, R. C. "Consecutive- k -out-of- n :F system with sequential failures." *IEEE Transactions on Reliability*, R-34 (Apr 1985): 43-45.
- [13] Bollinger, R. C. "Strict consecutive- k -out-of- n :F systems." *IEEE Transactions on Reliability*, R-34 (Apr 1985): 50-52.
- [14] Bollinger, R. C. "An algorithm for direct computation in consecutive- k -out-of- n :F system." *IEEE Transactions on Reliability*, R-35 (Dec 1986): 611-612.
- [15] Bollinger, R. C. and A. A. Salvia. "Consecutive- k -out-of- n :F networks." *IEEE Transactions on Reliability*, R-31 (Apr 1982): 53-55.
- [16] Chao, M. T. and G. D. Lin. "Economical design of large consecutive- k -out-of- n :F systems." *IEEE Transactions on Reliability*, R-33 (Dec 1984): 411-413.
- [17] Chen, R. W. and F. K. Hwang. "Failure distributions of consecutive- k -out-of- n :F systems." *IEEE Transactions on Reliability*, R-34 (Oct 1985): 338-341.
- [18] Chiang, D. T. and R. F. Chiang. "Relayed communication via consecutive- k -out-of- n :F system." *IEEE Transactions on Reliability*, R-35 (Apr 1986): 65-67.
- [19] Chiang, D. T. and S. C. Niu. "Reliability of consecutive- k -out-of- n :F system." *IEEE Transactions on Reliability*, R-30 (Apr 1981): 87-89.
- [20] Chung, K. L. "On the probability of the occurrence of at least m events among n arbitrary events." *Annals of Mathematical Statistics*, 12 (1941): 328-338.
- [21] Comtet, L. *Advanced Combinatorics*. D. Reidel, Boston, 1974.
- [22] Derman, C., G. J. Lieberman, and S. M. Ross. "On the consecutive- k -out-of- n :F system." *IEEE Transactions on Reliability*, R-31 (Apr 1982): 57-63.
- [23] Du, D. Z. and F. K. Hwang. "Optimal consecutive-2 systems of lines and cycles." *Networks*, 15 (1985): 439-447.
- [24] Du, D. Z. and F. K. Hwang. "Optimal consecutive-2-out-of- n systems." *Mathematics of Operations Research*, 11 (Feb 1986): 187-191.
- [25] Du, D. Z. and F. K. Hwang. "Optimal assignments for consecutive-2 graphs." *SIAM Journal on Algebraic and Discrete Methods*, 8 (Jul 1987): 510-518.
- [26] Du, D. Z. and F. K. Hwang. "A direct algorithm for computing reliability of a consecutive- k cycle." *IEEE Transactions on Reliability*, R-37 (Apr 1988): 70-72.
- [27] Feller, W. *An Introduction to Probability Theory and Its Applications*. Volume 1. 3rd edition. John Wiley & Sons, New York, 1968.

- [28] Franklin, G. F., J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, Reading, MA, 1986.
- [29] Fratta, L. and U. G. Montanari. "A Boolean algebra method for computing the terminal reliability in a communication network." *IEEE Transactions on Circuit Theory*, CT-20 (May 1973): 203-211.
- [30] Fu, J. C. "Reliability of a large consecutive- k -out-of- n :F system." *IEEE Transactions on Reliability*, R-34 (Jun 1985): 127-130.
- [31] Fu, J. C. "Bounds for reliability of large consecutive- k -out-of- n :F systems with unequal component reliability." *IEEE Transactions on Reliability*, R-35 (Aug 1986): 316-319.
- [32] Fu, J. C. "Reliability of consecutive- k -out-of- n :F systems with $(k - 1)$ -step Markov dependence." *IEEE Transactions on Reliability*, R-35 (Dec 1986): 602-606.
- [33] Fu, J. C. and Beihua Hu. "On reliability of a large consecutive- k -out-of- n :F system with $(k - 1)$ -step Markov dependence." *IEEE Transactions on Reliability*, R-36 (Apr 1987): 75-77.
- [34] Fussell, J. B. "How to hand-calculate system reliability and safety characteristics." *IEEE Transactions on Reliability*, R-24 (Aug 1975): 169-174.
- [35] Griffith, W. S. "On consecutive- k -out-of- n failure systems and their generalizations." In *Reliability and Quality Control*, ed. Asit P. Basu, 157-165. Elsevier Science Publishers B. V., North-Holland, 1986.
- [36] Griffith, W. S. and Z. Govindarajulu. "Consecutive- k -out-of- n Failure systems: reliability, availability, component importance, and multistate extensions." *American Journal of Mathematical and Management Sciences*, 5, Nos 1 & 2 (1985): 125-160.
- [37] Heidtmann, K. D. "Improved method of inclusion-exclusion applied to k -out-of- n systems." *IEEE Transactions on Reliability*, R-31 (Apr 1982): 36-39.
- [38] Heidtmann, K. D. Author reply, *IEEE Transactions on Reliability*, R-33 (Oct 1984): 322.
- [39] Hillier, F. S. and Gerald J. Lieberman. *Operations Research*. 2nd edition. Holden-Day, San Francisco, 1974.
- [40] Hwang, F. K. "Fast solutions for consecutive- k -out-of- n :F system." *IEEE Transactions on Reliability*, R-31 (Dec 1982): 447-448.

- [41] Hwang, F. K. "Relayed consecutive- k -out-of- $n:F$ lines." *IEEE Transactions on Reliability*, R-37 (Dec 1988): 512-514.
- [42] Hwang, F. K. "Invariant permutations for consecutive- k -out-of- n cycles." *IEEE Transactions on Reliability*, R-38 (Apr 1989): 65-67.
- [43] Jain, S. P. and Krishna Gopal. "Recursive algorithm for reliability evaluation of k -out-of- $n:G$ system." *IEEE Transactions on Reliability*, R-34 (Jun 1985): 144-146.
- [44] Kao, S. C. "Computing reliability from warranty." *Proceedings of the Statistical Computing Section, American Statistical Association*, 1982: 309-312.
- [45] Kontoleon, J. M. "Reliability determination of a r -successive-out-of- $n:F$ system." *IEEE Transactions on Reliability*, R-29 (Dec 1980): 437.
- [46] Kossow, A. and W. Preuss. "Failure probability of strict consecutive- k -out-of- $n:F$ systems." *IEEE Transactions on Reliability*, R-36 (Dec 1987): 551-553.
- [47] Kuo, B. C. *Automatic Control Systems*. 4th edition. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [48] Kuo, W. and W. Zhang. "Comments and inferences on: the most important components in a consecutive- k -out-of- $n:F$ system." Submitted to *IEEE Transactions on Reliability*, 1987.
- [49] Kuo, W., M. Zuo, and W. Zhang. "Consecutive- k -out-of- $n:G$ system: the mirror image of consecutive- k -out-of- $n:F$ system." Submitted to *IEEE Transactions on Reliability*, 1988.
- [50] Lambert, H. E. "Measures of importance of events and cut sets in fault trees." *Reliability and Fault Tree Analysis*, eds. R. E. Barlow, J. B. Fussel, and N. D. Singpurwalla, 77-100. Society for Industrial and Applied Mathematics, Philadelphia, 1975.
- [51] Lambiris, M. and S. G. Papastavridis. "Exact reliability formulas for linear and circular consecutive- k -out-of- $n:F$ systems." *IEEE Transactions on Reliability*, R-34 (Jun 1985): 124-126.
- [52] Locks, M. O. "Recursive disjoint products, inclusion-exclusion, and min-cut approximations." *IEEE Transactions on Reliability*, R-29 (Dec 1980): 368-371.
- [53] Locks, M. O. "Recursive disjoint products: a review of three algorithms." *IEEE Transactions on Reliability*, R-31 (Apr 1982): 33-35.

- [54] Locks, M. O. "Recent developments in computing of system reliability." *IEEE Transactions on Reliability*, R-34 (Dec 1985): 425-436.
- [55] Locks, M. O. "A minimizing algorithm for sum of disjoint products." *IEEE Transactions on Reliability*, R-36 (Oct 1987): 445-453.
- [56] Malon, D. M. "Optimal consecutive-2-out-of- $n:F$ component sequencing." *IEEE Transactions on Reliability*, R-33 (Dec 1984): 414-418.
- [57] Malon, D. M. "Optimal consecutive- k -out-of- $n:F$ component sequencing." *IEEE Transactions on Reliability*, R-34 (Apr 1985): 46-49.
- [58] Natvig, B. "On the reduction in remaining system lifetime due to the failure of a specific component." *Journal of Applied Probability*, 19 (1982): 642-652.
- [59] Olmstead, P. S. "Note on theoretical and observed distributions of repetitive occurrences." *Annals of Mathematical Statistics*, 11 (1940): 363-366.
- [60] Pan, Zhi-Jie and Ya-Chuan Tai. "Variance importance of system components by Monte Carlo." *IEEE Transactions on Reliability*, R-37 (Oct 1988): 421-423.
- [61] Papastavridis, S. G. "The most important component in a consecutive- k -out-of- $n:F$ system." *IEEE Transactions on Reliability*, R-36 (Jun 1987): 266-268.
- [62] Papastavridis, S. G. "Upper and lower bounds for the reliability of a consecutive- k -out-of- $n:F$ system." *IEEE Transactions on Reliability*, R-35 (Dec 1986): 607-610.
- [63] Papastavridis, S. G. "A limit theorem for the reliability of a consecutive- k -out-of- n system." *Advances in Applied Probability*, 19 (1987): 746-748.
- [64] Papastavridis, S. G. and J. Hadjichristos. "Mean time to failure for a consecutive- k -out-of- $n:F$ system." *IEEE Transactions on Reliability*, R-36 (Apr 1987): 85-86.
- [65] Papastavridis, S. G. and J. Hadjichristos. "Formulas for the reliability of a consecutive- k -out-of- $n:F$ system." *Journal of Applied Probability*, 26 (1988): 772-779.
- [66] Papastavridis, S. G. and M. Lambiris. "Reliability of a consecutive- k -out-of- $n:F$ system for Markov-dependent components." *IEEE Transactions on Reliability*, R-36 (Apr 1987): 78-79.
- [67] Rai, S., A. K. Sarje, E. V. Prasad, & Arun Kumar. "Two recursive algorithms for computing the reliability of k -out-of- n systems." *IEEE Transactions on Reliability*, R-36 (Jun 1987): 261-265.

- [68] Rushdi, A. M. "Symbolic reliability analysis with the aid of variable-entered Karnaugh maps." *IEEE Transactions on Reliability*, R-32 (Jun 1983): 134-139.
- [69] Rushdi, A. M. "How to hand-check a symbolic reliability expression." *IEEE Transactions on Reliability*, R-32 (1983): 402-408.
- [70] Rushdi, A. M. "Utilization of symmetric switching functions in the computation of k -out-of- n system reliability." *Microelectronics and Reliability*, 26 (1986): 973-987.
- [71] Salvia, A. A. "Simple inequalities for consecutive- k -out-of- $n:F$ networks." *IEEE Transactions on Reliability*, R-31 (Dec 1982): 450.
- [72] Scheuer, E. M. "Reliability of an m -out-of- n system when component failure induces higher failure rates in survivors." *IEEE Transactions on Reliability*, R-37 (Apr 1988): 73-74.
- [73] Shanthikumar, J. G. "Recursive algorithm to evaluate the reliability of a consecutive- k -out-of- $n:F$ system." *IEEE Transactions on Reliability*, R-31 (Dec 1982): 442-443.
- [74] Shanthikumar, J. G. "Lifetime distribution of consecutive- k -out-of- $n:F$ systems with exchangeable lifetimes." *IEEE Transactions on Reliability*, R-34 (Dec 1985): 480-483.
- [75] Shih, S. C. "Poisson limit of reliability in consecutive- k -out-of- $n:F$ systems." *Technical Paper*, Institute of Statistics, Academia Sinica, ROC, 1985.
- [76] Tong, Y. L. "A rearrangement inequality for the longest run, with an application to network reliability." *Journal of Applied Probability*, 22 (1985): 386-393.
- [77] Tong, Y. L. "Some new results on the reliability of circular consecutive- k -out-of- $n:F$ system." In *Reliability and Quality Control*, ed. A. P. Basu, 395-400. Elsevier Science Publishers B. V., North-Holland, 1986.
- [78] Wei, V. K., F. K. Hwang, and V. T. Sos. "Optimal sequencing of items in a consecutive-2-out-of- n system." *IEEE Transactions on Reliability*, R-32 (Apr 1983): 30-33.
- [79] Zhang, W. unpublished Doctoral dissertation. Parks Library, Iowa State University, Ames, Iowa, 1988.
- [80] Zuo, Mingjian and Way Kuo. "Reliability design for high system utilization." In *Advances in Reliability and Quality Control*, ed. M. H. Hamza, 53-56. ACTA Press, Anaheim, Canada, 1988.



1951

1952

1953

1954

1955

1956

